IBM

Application System/400™

Binder SX21-9925-0

Digitized by the Internet Archive
in 2026

https://archive.org/details/manualsplus_12446

IBM

AS/400™

SC21-9779-1

Programming:
Control Language Reference
Volume 5

IBM

AS/400™

SC21-9779-1

Programming:
Control Language Reference
Volume 5

Second Edition (October 1988)

This major revision makes obsolete SC21-9779-0. Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change or addition.

This edition applies to Release 1, Modification Level 2 of the IBM Operating System/400 Licensed Program (Program 5728-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

Changes are periodically made to the information herein; any such changes will be reported in subsequent revisions or technical newsletters.

This publication contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent program may be used instead.

The numbers at the bottom right of illustrations are publishing control numbers and are not part of the technical content of this manual.

Publications are not stocked at the address given below. Requests for IBM publications should be made to your IBM representative or to your IBM-approved remarketer.

This publication could contain technical inaccuracies or typographical errors. A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Information Development, Department 245, Rochester, Minnesota, U.S.A. 55901. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

AS/400, Operating System/400, and OS/400 are trademarks of the International Business Machines Corporation.

© Copyright International Business Machines Corporation 1988

About This Manual

This volume describes the AS/400 control language (CL) commands and shows their syntax from command RNMM (Rename Member) through WRKWTR (Work with Writers).

How This Manual is Organized

The *CL Reference* is divided into five volumes: Volume 1 contains five chapters and nine appendixes (A through I) and Volumes 2 through 5 contain the command descriptions. This volume is Volume 5.

Volume 1 provides an overview of the CL commands and describes the syntax coding rules needed to code them.

Volumes 2 through 5 describe every control language command, including commands for the OS/400 licensed program and commands for languages and utilities.

Volume 2 contains commands ADDACC through CPYDOC.

Volume 3 contains commands CPYF through DLTPGM.

Volume 4 contains commands DLTPNLGRP through RNMDLO.

Volume 5 contains commands RNMM through WRKWTR.

What You Should Know

An “About This Manual” is located at the front of each volume to help you know what information is in each volume.

A table of contents is placed in each volume. The table of contents for Volume 1 lists the headings in that volume, as well as the main headings of the remaining volumes. The table of contents for Volumes 2 through 5 lists the main heading for Volume 1, as well as the command name headings for Volumes 2 through 5.

The CL commands are described in alphabetical order in Volumes 2 through 5.

The glossary, located in the back of Volume 1, defines words used in this manual. Use the glossary if you find an unfamiliar word.

The index, located in the back of Volume 1, applies only to Volume 1. For Volumes 2 through 5, use the table of contents to find a particular command.

Contents

Chapter 1. Summary of OS/400 Functions and Object Types V1-1

Chapter 2. Control Language Syntax V1-25

Chapter 3. Format of Command Descriptions V1-55

Chapter 4. Command Definition Statements V1-69

Appendix A. Expanded Parameter Descriptions V1-107

Appendix B. Expressions V1-139

Appendix C. User Profile Matrix Chart V1-149

Appendix D. Files Used by CL Commands V1-153

Appendix E. Error Messages That Can Be Monitored V1-163

Appendix F. Command and Keyword Abbreviations V1-219

Appendix G. Generic Functions V1-223

Appendix H. Parameter Values Used for Testing and Debugging V1-225

Appendix I. Using Double-Byte Character Text in CL Commands V1-229

Glossary V1-233

Index V1-267

Command Descriptions ADDACC through CPYDOC V2-1
ADDACC (Add Access Code) Command V2-1
ADDAJE (Add Autostart Job Entry) Command V2-3
ADDAUTLE (Add Authorization List Entry) Command V2-5
ADDBKP (Add Breakpoint) Command V2-7
ADDCFGLE (Add Configuration List Entries) Command V2-11
ADDCMNE (Add Communications Entry) Command V2-14
ADDDIRE (Add Directory Entry) Command V2-17
ADDDLOAUT (Add Document Library Object Authority) Command V2-21
ADDDSTLE (Add Distribution List Entry) Command V2-23
ADDDTADFN (Add Data Definition) Command V2-25
ADDFCTE (Add Forms Control Table Entry) Command V2-27
ADDICFDEVE (Add Intersystem Communications Function Program Device Entry) Command V2-32
ADDJOBQE (Add Job Queue Entry) Command V2-38
ADDLFM (Add Logical File Member) Command V2-40
ADDLIBLE (Add Library List Entry) Command V2-43
ADDMSGD (Add Message Description) Command V2-44
ADDNETJOBE (Add Network Job Entry) Command V2-53
ADDPFM (Add Physical File Member) Command V2-56
| ADDPFVLM (Add Physical File Variable-Length Member) Command V2-58
ADDPGM (Add Program) Command V2-59
ADDRJECMNE (Add RJE Communications Entry) Command V2-60
ADDRJERDRE (Add RJE Reader Entry) Command V2-62
ADDRJEWTRE (Add RJE Writer Entry) Command V2-64
ADDRPYLE (Add Reply List Entry) Command V2-68
ADDRTGE (Add Routing Entry) Command V2-72
ADDTRC (Add Trace) Command V2-75

ADDWSE (Add Work Station Entry) Command V2-78
ALCOBJ (Allocate Object) Command V2-81
ANSLIN (Answer Line) Command V2-84
ANSQST (Answer Questions) Command V2-85
ANZACCGRP (Analyze Access Group) Command V2-86
ANZDBF (Analyze Data Base File) Command V2-88
ANZDBFKEY (Analyze Data Base File Keys) Command V2-89
ANZPGM (Analyze Program) Command V2-90
ANZPRB (Analyze Problem) Command V2-91
APYJRNCHG (Apply Journaled Changes) Command V2-92
APYPTF (Apply Program Temporary Fix) Command V2-96
ASKQST (Ask Question) Command V2-98
BCHJOB (Batch Job) Command V2-99
CALL (Call Program) Command V2-105
CFGDSTSRV (Configure Distribution Services) Command V2-108
CFGRPDS (Configure RSCS/PROFS Distribution Services) Command V2-110
CHGACGCDE (Change Accounting Code) Command V2-111
CHGAJE (Change Autostart Job Entry) Command V2-112
CHGAUTLE (Change Authorization List Entry) Command V2-114
CHGCFGL (Change Configuration List) Command V2-116
CHGCMD (Change Command) Command V2-120
CHGCMDDFT (Change Command Default) Command V2-123
CHGCMNE (Change Communications Entry) Command V2-126
CHGCOSD (Change Class-of-Service Description) Command V2-129
CHGCTLAPPC (Change Controller Description (APPC)) Command V2-133
CHGCTLASC (Change Controller Description (Async)) Command V2-141
CHGCTLBSC (Change Controller Description (BSC)) Command V2-145
CHGCTLFNC (Change Controller Description (Finance)) Command V2-147
CHGCTLHOST (Change Controller Description (SNA Host)) Command V2-152
CHGCTLLWS (Change Controller Description (Local Work Station)) Command V2-160
CHGCTLRWS (Change Controller Description (Remote Work Station)) Command V2-162
CHGCTLVWS (Change Controller Description (Virtual Work Station)) Command V2-169
CHGCURLIB (Change Current Library) Command V2-170
CHGDBG (Change Debug) Command V2-171
CHGDDMF (Change Distributed Data Management File) Command V2-173
CHGDEVAPPC (Change Device Description (APPC)) Command V2-178
CHGDEVASC (Change Device Description (Async)) Command V2-180
CHGDEVBSC (Change Device Description (BSC)) Command V2-181
CHGDEVDKT (Change Device Description (Diskette)) Command V2-185
CHGDEVDSP (Change Device Description (Display)) Command V2-186
CHGDEVFNC (Change Device Description for (Finance)) Command V2-194
CHGDEVHOST (Change Device Description (SNA Host)) Command V2-195
CHGDEVPRT (Change Device Description (Printer)) Command V2-197
CHGDEVSNUF (Change Device Description (SNUF)) Command V2-205
CHGDEVTAP (Change Device Description (Tape)) Command V2-207
CHGDIRE (Change Directory Entry) Command V2-208
CHGDKTF (Change Diskette File) Command V2-211
CHGDLOAUT (Change Document Library Object Authority) Command V2-215
CHGDLOOWN (Change Document Library Object Owner) Command V2-218
CHGDOCD (Change Document Description) Command V2-220
CHGDSPF (Change Display File) Command V2-226
CHGDSTPWD (Change Dedicated Service Tools Password) Command V2-230
CHGDTA (Change Data) Command V2-231
CHGDTAARA (Change Data Area) Command V2-232
CHGFCT (Change Forms Control Table) Command V2-234
CHGFCTE (Change Forms Control Table Entry) Command V2-235
CHGGRPA (Change Group Attributes) Command V2-240
CHGHLLPTR (Change High-Level Language Pointer) Command V2-242
CHGICFDEVE (Change Intersystem Communications Function Program Device Entry) Command V2-245
CHGICFF (Change Intersystem Communications Function File) Command V2-252
CHGJOB (Change Job) Command V2-254

CHGJOBD (Change Job Description) Command V2-261
CHGJOBQE (Change Job Queue Entry) Command V2-268
CHGJRN (Change Journal) Command V2-270
CHGKBDMAP (Change Keyboard Map) Command V2-273
CHGLF (Change Logical File) Command V2-275
CHGLFM (Change Logical File Member) Command V2-279
CHGLIB (Change Library) Command V2-281
CHGLIBL (Change Library List) Command V2-282
CHGLINASC (Change Line Description (Async)) Command V2-283
CHGLINBSC (Change Line Description (BSC)) Command V2-288
CHGLINSDLC (Change Line Description (SDLC)) Command V2-293
CHGLINTDLC (Change Line Description (TDLC)) Command V2-300
CHGLINTRN (Change Line Description (Token-Ring Network)) Command V2-301
CHGLINX25 (Change Line Description (X.25)) Command V2-305
CHGMNU (Change Menu) Command V2-310
CHGMODD (Change Mode Description) Command V2-313
CHGMSGD (Change Message Description) Command V2-315
CHGMSGQ (Change Message Queue) Command V2-323
CHGNETA (Change Network Attributes) Command V2-326
CHGNETJOBE (Change Network Job Entry) Command V2-333
CHGOBJD (Change Object Description) Command V2-336
CHGOBJOWN (Change Object Owner) Command V2-337
CHGOUTQ (Change Output Queue) Command V2-339
CHGPCOPRF (Change PC Support Organizer Profile) Command V2-341
CHGPF (Change Physical File) Command V2-342
CHGPFM (Change Physical File Member) Command V2-347
CHGPGM (Change Program) Command V2-349
CHGPGMVAR (Change Program Variable) Command V2-351
CHGPRF (Change Profile) Command V2-354
CHGPRTF (Change Printer File) Command V2-359
CHGPTR (Change Pointer) Command V2-373
CHGPWD (Change Password) Command V2-376
CHGQSTDB (Change Question-and-Answer Data Base) Command V2-377
CHGRJECMNE (Change RJE Communications Entry) Command V2-378
CHGRJERDRE (Change RJE Reader Entry) Command V2-380
CHGRJEWTRE (Change RJE Writer Entry) Command V2-382
CHGRPYLE (Change Reply List Entry) Command V2-385
CHGRTGE (Change Routing Entry) Command V2-388
CHGSAVF (Change Save File) Command V2-391
CHGSBSD (Change Subsystem Description) Command V2-393
CHGSPLFA (Change Spooled File Attributes) Command V2-396
CHGSRCPF (Change Source Physical File) Command V2-401
CHGSSND (Change Session Description) Command V2-406
CHGSSNMAX (Change Session Maximum) Command V2-408
CHGSYSLIBL (Change System Library List) Command V2-410
CHGSYSVAL (Change System Value) Command V2-411
CHGS36 (Change System/36) Command V2-412
CHGS36PGMA (Change System/36 Program Attributes) Command V2-413
CHGS36PRCA (Change System/36 Procedure Attributes) Command V2-414
CHGS36SRCA (Change System/36 Source Attributes) Command V2-416
CHGTAPF (Change Tape File) Command V2-418
CHGUSRPRF (Change User Profile) Command V2-427
CHGVAR (Change Variable) Command V2-435
CHGWSE (Change Work Station Entry) Command V2-439
CHGWTR (Change Writer) Command V2-441
CHKDKT (Check Diskette) Command V2-443
CHKDOC (Check Document) Command V2-445
CHKIGCTBL (Check DBCS Table) Command V2-446
CHKOBJ (Check Object) Command V2-447
CHKPWD (Check Password) Command V2-450
CHKRCDLCK (Check Record Locks) Command V2-451
CHKTAP (Check Tape) Command V2-452
CLOF (Close File) Command V2-455

CLRDKT (Clear Diskette) Command V2-456
CLRJOBQ (Clear Job Queue) Command V2-458
CLRLIB (Clear Library) Command V2-459
CLRMSGQ (Clear Message Queue) Command V2-460
CLROUTQ (Clear Output Queue) Command V2-461
CLRPFM (Clear Physical File Member) Command V2-462
CLRSAVF (Clear Save File) Command V2-463
CLRTRCDTA (Clear Trace Data) Command V2-464
CMPJRNIMG (Compare Journal Images) Command V2-465
CNLRJERDR (Cancel RJE Reader) Command V2-469
CNLRJEWTR (Cancel RJE Writer) Command V2-470
COMMIT (Commit) Command V2-471
CPYCFGL (Copy Configuration List) Command V2-472
CPYDOC (Copy Document) Command V2-473

Command Descriptions CPYF through DLTPGM V3-1
CPYF (Copy File) Command V3-1
CPYFRMDKT (Copy from Diskette) Command V3-20
CPYFRMPCD (Copy from PC Document) Command V3-24
CPYFRMTAP (Copy from Tape) Command V3-27
CPYIGCSRT (Copy IGC Sort) Command V3-33
CPYIGCTBL (Copy DBCS Table) Command V3-34
CPYLIB (Copy Library) Command V3-39
CPYPTF (Copy Program Temporary Fix) Command V3-40
CPYSPLF (Copy Spooled File) Command V3-43
CPYSRCF (Copy Source File) Command V3-47
CPYTODKT (Copy to Diskette) Command V3-50
CPYTOPCD (Copy to PC Document) Command V3-53
CPYTOTAP (Copy to Tape) Command V3-56
CRTAPAR (Create Authorized Program Analysis Report) Command V3-61
CRTAUTHLR (Create Authority Holder) Command V3-66
CRTAUTL (Create Authorization List) Command V3-68
CRTBASPGM (Create Basic Program) Command V3-70
CRTCBLPGM (Create COBOL Program) Command V3-73
CRTCFGL (Create Configuration List) Command V3-77
CRTCLPGM (Create Control Language Program) Command V3-81
CRTCLS (Create Class) Command V3-84
CRTCMD (Create Command) Command V3-87
CRTCOSD (Create Class-of-Service Description) Command V3-92
CRTCTLAPPC (Create Controller Description (APPC)) Command V3-97
CRTCTLASC (Create Controller Description (Async)) Command V3-107
CRTCTLBSC (Create Controller Description (BSC)) Command V3-111
CRTCTLFNC (Create Controller Description (Finance)) Command V3-114
CRTCTLHOST (Create Controller Description (SNA Host)) Command V3-120
CRTCTLLWS (Create Controller Description (Local Work Station)) Command V3-127
CRTCTLRWS (Create Controller Description (Remote Work Station)) Command V3-129
CRTCTLVWS (Create Controller Description (Virtual Work Station)) Command V3-137
CRTDDMF (Create Distributed Data Management File) Command V3-138
CRTDEVAPPC (Create Device Description (APPC)) Command V3-144
CRTDEVASC (Create Device Description (Async)) Command V3-147
CRTDEVBSC (Create Device Description (BSC)) Command V3-148
CRTDEVDKT (Create Device Description (Diskette)) Command V3-153
CRTDEVDSP (Create Device Description (Display)) Command V3-155
CRTDEVFNC (Create Device Description (Finance)) Command V3-164
CRTDEVHOST (Create Device Description (SNA Host)) Command V3-166
CRTDEVPRT (Create Device Description (Printer)) Command V3-168
CRTDEVSNUF (Create Device Description (SNUF)) Command V3-177
CRTDEVTAP (Create Device Description (Tape)) Command V3-179
CRTDKTF (Create Diskette File) Command V3-181
CRTDOC (Create Document) Command V3-186
CRTDSPF (Create Display File) Command V3-187
CRTDSTL (Create Distribution List) Command V3-193
CRTDTAARA (Create Data Area) Command V3-194

CRTDTADCT (Create a Data Dictionary) Command V3-196
CRTDTAQ (Create Data Queue) Command V3-197
CRTDUPOBJ (Create Duplicate Object) Command V3-199
CRTEDTD (Create Edit Description) Command V3-202
CRTFCT (Create Forms Control Table) Command V3-207
CRTFLR (Create Folder) Command V3-208
CRTGSS (Create Graphics Symbol Set) Command V3-211
CRTICFF (Create Intersystem Communications Function File) Command V3-213
CRTIGCDCT (Create DBCS Conversion Dictionary) Command V3-217
CRTJOBD (Create Job Description) Command V3-218
CRTJOBQ (Create Job Queue) Command V3-225
CRTJRN (Create Journal) Command V3-227
CRTJRNRCV (Create Journal Receiver) Command V3-229
CRTLF (Create Logical File) Command V3-231
CRTLIB (Create Library) Command V3-240
CRTLINASC (Create Line Description (Async)) Command V3-242
CRTLINBSC (Create Line Description (BSC)) Command V3-247
CRTLINSDLC (Create Line Description (SDLC)) Command V3-252
CRTLINTDLC (Create Line Description (TDLC)) Command V3-260
CRTLINTRN (Create Line Description (Token-Ring Network)) Command V3-261
CRTLINX25 (Create Line Description (X.25)) Command V3-265
CRTMNU (Create Menu) Command V3-271
CRTMODD (Create Mode Description) Command V3-274
CRTMSGF (Create Message File) Command V3-276
CRTMSGFMNU (Create Message File Menu) Command V3-278
CRTMSGQ (Create Message Queue) Command V3-281
CRTOUTQ (Create Output Queue) Command V3-283
CRTPASPGM (Create Pascal Program) Command V3-285
CRTPF (Create Physical File) Command V3-291
CRTPLIPGM (Create PL/I Language Program) Command V3-300
CRTPRTF (Create Printer File) Command V3-305
CRTQSTDB (Create Question and Answer Data Base) Command V3-326
CRTQSTLOD (Create Question-and-Answer Load) Command V3-327
CRTRJEBSCF (Create RJE BSC File) Command V3-328
CRTRJECFG (Create RJE Configuration) Command V3-331
CRTRJECMNF (Create RJE Communications File) Command V3-338
CRTRPGPGM (Create RPG/400 Program) Command V3-341
CRTRPTPGM (Create Auto Report Program) Command V3-345
CRTSAVF (Create Save File) Command V3-350
CRTSBSD (Create Subsystem Description) Command V3-352
CRTSPADCT (Create Spelling Aid Dictionary) Command V3-355
CRTSQLCBL (Create Structured Query Language COBOL) Command V3-359
CRTSQLPLI (Create Structured Query Language PL/I) Command V3-362
CRTSQLRPG (Create Structured Query Language RPG) Command V3-365
CRTSRCPF (Create Source Physical File) Command V3-368
CRTSSND (Create Session Description) Command V3-375
CRTS36CBL (Create System/36 COBOL) Command V3-378
CRTS36DSPF (Create System/36 Display File) Command V3-383
CRTS36MNU (Create System/36 Menu) Command V3-386
CRTS36MSGF (Create System/36 Message File) Command V3-389
CRTS36RPG (Create System/36 RPG) Command V3-394
CRTS36RPGR (Create System/36 RPGR) Command V3-398
CRTS36RPT (Create System/36 RPT) Command V3-400
CRTTAPF (Create Tape File) Command V3-405
CRTTBL (Create Table) Command V3-415
CRTUSRPRF (Create User Profile) Command V3-417
CVTCLSRC (Convert CL Source) Command V3-426
CVTDAT (Convert Date) Command V3-428
CVTRJEDTA (Convert RJE Data) Command V3-430
CVTTOFLR (Convert to Folder) Command V3-433
DATA (Data) Command V3-434
DCL (Declare CL Variable) Command V3-436
DCLF (Declare File) Command V3-438

DLCOBJ (Deallocate Object) Command V3-441
DLTAUTHLR (Delete Authority Holder) Command V3-443
DLTAUTL (Delete Authorization List) Command V3-444
DLTCFGL (Delete Configuration List) Command V3-445
DLTCHTFMT (Delete Chart Format) Command V3-446
DLTCLS (Delete Class) Command V3-447
DLTCMD (Delete Command) Command V3-448
DLTCOSD (Delete Class-of Service Description) Command V3-449
DLTCTLD (Delete Controller Description) Command V3-450
DLTDEVD (Delete Device Description) Command V3-451
DLTDFUPGM (Delete DFU Program) Command V3-452
DLTDKTLBL (Delete Diskette Label) Command V3-453
DLTDLO (Delete Document Library Object) Command V3-455
DLTDOCL (Delete Document List) Command V3-459
DLTDST (Delete Distribution) Command V3-461
DLTDSTL (Delete Distribution List) Command V3-463
DLTDTAARA (Delete Data Area) Command V3-464
DLTDTADCT (Delete Data Dictionary) Command V3-465
DLTDTAQ (Delete Data Queue) Command V3-466
DLTEDTD (Delete Edit Description) Command V3-467
DLTF (Delete File) Command V3-468
DLTFCT (Delete Forms Control Table) Command V3-470
DLTGSS (Delete Graphics Symbol Set) Command V3-471
DLTIGCDCT (Delete DBCS Conversion Dictionary) Command V3-472
DLTIGCSRT (Delete IGC Sort) Command V3-473
DLTIGCTBL (Delete DBCS Table) Command V3-474
DLTJOBD (Delete Job Description) Command V3-476
DLTJOBQ (Delete Job Queue) Command V3-477
DLTJRN (Delete Journal) Command V3-478
DLTJRNRCV (Delete Journal Receiver) Command V3-479
DLTLIB (Delete Library) Command V3-481
DLTLICPGM (Delete Licensed Program) Command V3-482
DLTLIND (Delete Line Description) Command V3-483
DLTMNU (Delete Menu) Command V3-484
DLTMODD (Delete Mode Description) Command V3-485
DLTMSGF (Delete Message File) Command V3-486
DLTMSGQ (Delete Message Queue) Command V3-487
DLTNETF (Delete Network File) Command V3-488
DLTOUTQ (Delete Output Queue) Command V3-490
DLTOVR (Delete Override) Command V3-491
DLTOVRDEVE (Delete Override Device Entry) Command V3-493
DLTPGM (Delete Program) Command V3-494

Command Descriptions DLTPNLGRP through RNMDLO V4-1
DLTPNLGRP (Delete Panel Group) Command V4-1
DLTPRB (Delete Problem) Command V4-3
DLTQRY (Delete Query) Command V4-4
DLTQST (Delete Question) Command V4-5
DLTQSTDB (Delete Question-and-Answer Data Base) Command V4-6
DLTRJECFG (Delete RJE Configuration) Command V4-7
DLTSBSD (Delete Subsystem Description) Command V4-8
DLTSCHIDX (Delete Search Index) Command V4-9
DLTSPADCT (Delete Spelling Aid Dictionary) Command V4-10
DLTSPLF (Delete Spooled File) Command V4-11
DLTSSND (Delete Session Description) Command V4-13
DLTTBL (Delete Table) Command V4-14
DLTUSRPRF (Delete User Profile) Command V4-15
DLYJOB (Delay Job) Command V4-16
DMPCLPGM (Dump CL Program) Command V4-17
DMPDLO (Dump Document Library Object) Command V4-18
DMPJOB (Dump Job) Command V4-18.2
DMPJOBINT (Dump Job Internal) Command V4-20
DMPOBJ (Dump Object) Command V4-21

DMPSYSOBJ (Dump System Object) Command V4-22
DMPTAP (Dump Tape) Command V4-26
DMPTRC (Dump Trace) Command V4-29
DO (Do) Command V4-30
DSPACC (Display Access Code) Command V4-31
DSPACCAUT (Display Access Code Authority) Command V4-32
DSPACCGRP (Display Access Group) Command V4-33
DSPAPPNINF (Display APPN Information) Command V4-35
DSPAUTHLR (Display Authority Holder) Command V4-37
DSPAUTL (Display Authorization List) Command V4-39
DSPAUTLDLO (Display Authorization List Document Library Objects) Command V4-41
DSPAUTLOBJ (Display Authorization List Objects) Command V4-42
DSPAUTUSR (Display Authorized Users) Command V4-44
DSPBKP (Display Breakpoints) Command V4-45
DSPCFGL (Display Configuration List) Command V4-46
DSPCHT (Display Chart) Command V4-47
DSPCLS (Display Class) Command V4-50
DSPCMD (Display Command) Command V4-51
DSPCOSD (Display Class-of-Service Description) Command V4-52
DSPCTLD (Display Controller Description) Command V4-53
DSPDBG (Display Debug) Command V4-54
DSPDBR (Display Data Base Relations) Command V4-55
DSPDDMF (Display Distributed Data Management File) Command V4-59
DSPDEVD (Display Device Description) Command V4-61
DSPDIR (Display Directory) Command V4-62
DSPDKT (Display Diskette) Command V4-65
DSPDLOAUT (Display Document Library Object Authority) Command V4-67
DSPDOC (Display Document) Command V4-69
DSPDSTL (Display Distribution List) Command V4-70
DSPDSTLOG (Display Distribution Log) Command V4-73
DSPDSTSRV (Display Distribution Services) Command V4-77
DSPDTA (Display Data) Command V4-79
DSPDTAARA (Display Data Area) Command V4-80
DSPDTADCT (Display Data Dictionary) Command V4-82
DSPEDTD (Display Edit Description) Command V4-84
DSPFD (Display File Description) Command V4-85
DSPFFD (Display File Field Description) Command V4-92
DSPFLR (Display Folder) Command V4-96
DSPGDF (Display Graphics Data File) Command V4-98
DSPHLPDOC (Display Help Document) Command V4-100
DSPIGCDCT (Display DBCS Conversion Dictionary) Command V4-101
DSPJOB (Display Job) Command V4-103
DSPJOBD (Display Job Description) Command V4-105
DSPJOBLOG (Display Job Log) Command V4-106
DSPJRN (Display Journal) Command V4-107
DSPJRNRCVA (Display Journal Receiver Attributes) Command V4-114
DSPKBDMAP (Display Keyboard Map) Command V4-115
DSPLIB (Display Library) Command V4-116
DSPLIBL (Display Library List) Command V4-118
DSPLIND (Display Line Description) Command V4-119
DSPLOG (Display Log) Command V4-120
DSPMNUA (Display Menu Attributes) Command V4-122
DSPMODD (Display Mode Description) Command V4-123
DSPMODSTS (Display Mode Status) Command V4-124
DSPMSG (Display Messages) Command V4-125
DSPMSGD (Display Message Descriptions) Command V4-127
DSPNETA (Display Network Attributes) Command V4-129
DSPOBJAUT (Display Object Authority) Command V4-130
DSPOBJD (Display Object Description) Command V4-132
DSPOVR (Display Override) Command V4-135
DSPPFM (Display Physical File Member) Command V4-137
DSPPGM (Display Program) Command V4-138
DSPPGMADP (Display Program Adopt) Command V4-139

DSPPGMREF (Display Program References) Command V4-141
DSPPGMVAR (Display Program Variable) Command V4-144
DSPPTF (Display Program Temporary Fix) Command V4-146
DSPRCDLCK (Display Record Locks) Command V4-148
DSPRJECFG (Display RJE Configuration) Command V4-149
DSPSAVF (Display Save File) Command V4-150
DSPSBSD (Display Subsystem Description) Command V4-151
DSPSOCSTS (Display Sphere of Control Status) Command V4-152
DSPSPLF (Display Spooled File) Command V4-153
DSPSRVSTS (Display Service Status) Command V4-154
DSPSYSVAL (Display System Value) Command V4-155
DSPS36 (Display System/36) Command V4-156
DSPTAP (Display Tape) Command V4-157
DSPTRC (Display Trace) Command V4-159
DSPTRCDTA (Display Trace Data) Command V4-160
DSPUSRPMN (Display User Permission) Command V4-161
DSPUSRPRF (Display User Profile) Command V4-162
DUPDKT (Duplicate Diskette) Command V4-164
EDTAUTL (Edit Authorization List) Command V4-166
EDTDLOAUT (Edit Document Library Object Authority) Command V4-167
EDTDOC (Edit Document) Command V4-168
EDTIGCDCT (Edit DBCS Conversion Dictionary) Command V4-169
EDTIGCTBL (Edit DBCS Table) Command V4-171
EDTLIBL (Edit Library List) Command V4-172
EDTOBJAUT (Edit Object Authority) Command V4-173
EDTQST (Edit Questions and Answers) Command V4-174
EDTS36PGMA (Edit System/36 Program Attributes) Command V4-175
EDTS36PRCA (Edit System/36 Procedure Attributes) Command V4-176
EDTS36SRCA (Edit System/36 Source Attributes) Command V4-177
EJTEMLOUT (Eject Emulation Output) Command V4-178
ELSE (Else) Command V4-180
EMLPRTKEY (Emulate Printer Key) Command V4-182
ENDBCHJOB (End Batch Job) Command V4-183
ENDCBLDBG (End COBOL Debug) Command V4-184
ENDCMTCTL (End Commitment Control) Command V4-185
ENDCPYSCN (End Copy Screen) Command V4-186
ENDCTLRCY (End Controller Recovery) Command V4-187
ENDDBG (End Debug) Command V4-188
ENDDEVRCY (End Device Recovery) Command V4-189
ENDDO (End Do) Command V4-190
ENDDSKCOL (End Disk Data Collection) Command V4-190.1
ENDGRPJOB (End Group Job) Command V4-191
ENDINP (End Input) Command V4-193
ENDJOB (End Job) Command V4-194
ENDJOBABN (End Job Abnormal) Command V4-196
ENDJOBTRC (End Job Trace) Command V4-198
ENDJRNAP (End Journaling Access Path) Command V4-200
ENDJRNPF (End Journaling Physical File Changes) Command V4-201
ENDLINRCY (End Line Recovery) Command V4-202
ENDMOD (End Mode) Command V4-203
ENDPASTHR (End Pass-Through) Command V4-205
ENDPFRMON (End Performance Monitor) Command V4-206
ENDPGM (End Program) Command V4-207
ENDPRTEML (End Printer Emulation) Command V4-208
ENDRCV (End Receive) Command V4-209
ENDRDR (End Reader) Command V4-210
ENDRJESSN (End RJE Session) Command V4-211
ENDRQS (End Request) Command V4-213
ENDSAM (End Sampled Address Monitor) Command V4-215
ENDSAMCOL (End Sampled Address Monitor Data Collection) Command . . V4-216
ENDSBS (End Subsystem) Command V4-218
ENDSRVJOB (End Service Job) Command V4-220
ENDSYS (End System) Command V4-221

ENDS36 (End System/36) Command V4-222
ENDTIESSN (End Technical Information Exchange Session) Command V4-223
ENDWTR (End Writer) Command V4-224
EXTPGMINF (Extract Program Information) Command V4-224.1
FILDOC (File Document) Command V4-225
FMTDTA (Format Data) Command V4-232
GO (Go to Menu) Command V4-234
GOTO (Go To) Command V4-236
GRTACCAUT (Grant Access Code Authority) Command V4-237
GRTOBJAUT (Grant Object Authority) Command V4-239
GRTUSRAUT (Grant User Authority) Command V4-243
GRTUSRPMN (Grant User Permission) Command V4-245
HLDCMNDEV (Hold Communications Device) Command V4-246
HLDJOB (Hold Job) Command V4-247
HLDJOBQ (Hold Job Queue) Command V4-248
HLDOUTQ (Hold Output Queue) Command V4-249
HLDRDR (Hold Reader) Command V4-250
HLDSPLF (Hold Spooled File) Command V4-251
HLDWTR (Hold Writer) Command V4-253
IF (If) Command V4-254
INZDKT (Initialize Diskette) Command V4-257
INZPCS (Initialize PC Support) Command V4-261
INZPFM (Initialize Physical File Member) Command V4-263
INZTAP (Initialize Tape) Command V4-265
LNKDTADFN (Link Data Definition) Command V4-268
LODPTF (Load Program Temporary Fix) Command V4-270
LODQSTDB (Load Question-and-Answer Data Base) Command V4-272
MDLSYS (Model System) Command V4-273
MONMSG (Monitor Message) Command V4-274
MOVOBJ (Move Object) Command V4-277
MRGDOC (Merge Document) Command V4-279
MRGFORMD (Merge Form Description) Command V4-281
MRGMSGF (Merge Message File) Command V4-284
OPNDBF (Open Data Base File) Command V4-286
OPNQRYF (Open Query File) Command V4-288
OVRDBF (Override with Data Base File) Command V4-310
OVRDKTF (Override with Diskette File) Command V4-318
OVRDSPF (Override with Display File) Command V4-323
OVRICFDEVE (Override Intersystem Communications Function Program Device Entry) Command V4-327
OVRICFF (Override with Intersystem Communications Function File) Command V4-334
OVRMSGF (Override with Message File) Command V4-337
OVRPRTF (Override with Printer File) Command V4-338
OVRSAVF (Override Save File) Command V4-352
OVRTAPF (Override with Tape File) Command V4-354
PAGDOC (Paginate Document) Command V4-363
PGM (Program) Command V4-364
POSDBF (Position Data Base File) Command V4-366
PRTCMDUSG (Print Command Usage) Command V4-367
PRTCPTRPT (Print Component Report) Command V4-369
PRTDEVADR (Print Device Addresses) Command V4-374
PRTDOC (Print Document) Command V4-374.1
PRTDSKRPT (Print Disk Activity Report) Command V4-375
PRTERRLOG (Print Error Log) Command V4-377
PRTINTDTA (Print Internal Data) Command V4-380
PRTJOBTRC (Print Job Trace) Command V4-382
PRTLCKRPT (Print Lock Report) Command V4-385
PRTSAMDTA (Print Sampled Address Monitor Data) Command V4-387
PRTSYSRPT (Print System Report) Command V4-389
PRTTNSRPT (Print Transaction Report) Command V4-394
PWRDWNSYS (Power Down System) Command V4-398
QRYDOCLIB (Query Document Library) Command V4-400
QRYDST (Query Distribution) Command V4-407

QRYTIEF (Query Technical Information Exchange File) Command V4-410
RCLDDMCNV (Reclaim Distributed Data Management Conversations) Command V4-411
RCLDLO (Reclaim Document Library Object) Command V4-412
RCLRSC (Reclaim Resources) Command V4-414
RCLSTG (Reclaim Storage) Command V4-417
RCVDST (Receive Distribution) Command V4-419
RCVF (Receive File) Command V4-423
RCVJRNE (Receive Journal Entry) Command V4-425
RCVMSG (Receive Message) Command V4-430
RCVNETF (Receive Network File) Command V4-436
RCVTIEF (Receive Technical Information Exchange File) Command V4-438
RETURN (Return) Command V4-439
RGZPFM (Reorganize Physical File Member) Command V4-440
RLSCMNDEV (Release Communications Device) Command V4-443
RLSJOB (Release Job) Command V4-444
RLSJOBQ (Release Job Queue) Command V4-445
RLSOUTQ (Release Output Queue) Command V4-446
RLSRDR (Release Reader) Command V4-447
RLSSPLF (Release Spooled File) Command V4-448
RLSWTR (Release Writer) Command V4-450
RMVACC (Remove Access Code) Command V4-452
RMVAJE (Remove Autostart Job Entry) Command V4-453
RMVAUTLE (Remove Authorization List Entry) Command V4-454
RMVBKP (Remove Breakpoint) Command V4-455
RMVCFGLE (Remove Configuration List Entries) Command V4-456
RMVCMNE (Remove Communications Entry) Command V4-458
RMVDIRE (Remove Directory Entry) Command V4-460
RMVDLOAUT (Remove Document Library Object Authority) Command . . . V4-462
RMVDSTLE (Remove Distribution List Entry) Command V4-464
RMVFCTE (Remove Forms Control Table Entry) Command V4-466
RMVICFDEVE (Remove Intersystem Communications Function Program Device Entry) Command V4-467
RMVJOBQE (Remove Job Queue Entry) Command V4-468
RMVJRNCHG (Remove Journaled Changes) Command V4-469
RMVLIBLE (Remove Library List Entry) Command V4-473
RMVM (Remove Member) Command V4-474
RMVMSG (Remove Message) Command V4-475
RMVMSGD (Remove Message Description) Command V4-477
RMVNETJOBE (Remove Network Job Entry) Command V4-478
RMVPGM (Remove Program) Command V4-479
RMVPTF (Remove Program Temporary Fix) Command V4-480
RMVRJECMNE (Remove RJE Communications Entry) Command V4-482
RMVRJERDRE (Remove RJE Reader Entry) Command V4-483
RMVRJEWTRE (Remove RJE Writer Entry) Command V4-484
RMVRPYLE (Remove Reply List Entry) Command V4-485
RMVRTGE (Remove Routing Entry) Command V4-486
RMVTRC (Remove Trace) Command V4-487
RMVWSE (Remove Work Station Entry) Command V4-489
RNMDKT (Rename Diskette) Command V4-490
RNMDLO (Rename Document Library Object) Command V4-491

Command Descriptions RNMM through WRKWTR V5-1
RNMM (Rename Member) Command V5-1
RNMOBJ (Rename Object) Command V5-3
ROLLBACK (Rollback) Command V5-5
RPLDOC (Replace Document) Command V5-6
RRTJOB (Reroute Job) Command V5-9
RSMBKP (Resume Breakpoint) Command V5-10
RSMCTLRCY (Resume Controller Recovery) Command V5-11
RSMDEVRCY (Resume Device Recovery) Command V5-12
RSMLINRCY (Resume Line Recovery) Command V5-13
RSTAUT (Restore Authority) Command V5-14

RSTCFG (Restore Configuration) Command V5-17
RSTDLO (Restore Document Library Object) Command V5-20
RSTLIB (Restore Library) Command V5-25
RSTLICPGM (Restore Licensed Program) Command V5-30
RSTOBJ (Restore Object) Command V5-33
RSTS36F (Restore System/36 File) Command V5-40
RSTS36FLR (Restore System/36 Folder) Command V5-45
RSTS36LIBM (Restore System/36 Library Members) Command V5-49
RSTUSRPRF (Restore User Profiles) Command V5-54
RTVAUTLE (Retrieve Authorization List Entry) Command V5-57
RTVCLSRC (Retrieve CL Source) Command V5-60
RTVDOC (Retrieve Document) Command V5-61
RTVDTAARA (Retrieve Data Area) Command V5-65
RTVGRPA (Retrieve Group Attributes) Command V5-68
RTVJOBA (Retrieve Job Attributes) Command V5-71
RTVJRNE (Retrieve Journal Entry) Command V5-76
RTVMSG (Retrieve Message) Command V5-82
RTVNETA (Retrieve Network Attributes) Command V5-85
RTVSYSVAL (Retrieve System Value) Command V5-90
RTVUSRPRF (Retrieve User Profile) Command V5-91
RUNQRY (Run Query) Command V5-95
RVKACCAUT (Revoke Access Code Authority) Command V5-101
RVKOBJAUT (Revoke Object Authority) Command V5-102
RVKUSRPMN (Revoke User Permission) Command V5-106
SAVCHGOBJ (Save Changed Object) Command V5-107
SAVDLO (Save Document Library Object) Command V5-113
SAVLIB (Save Library) Command V5-119
SAVLICPGM (Save Licensed Program) Command V5-124
SAVOBJ (Save Object) Command V5-127
SAVSAVFDTA (Save Save File Data) Command V5-134
SAVSYS (Save System) Command V5-137
SAVS36F (Save System/36 File) Command V5-139
SAVS36LIBM (Save System/36 Library Members) Command V5-143
SBMDBJOB (Submit Data Base Jobs) Command V5-147
SBMDKTJOB (Submit Diskette Jobs) Command V5-149
SBMFNCJOB (Submit Finance Job) Command V5-152
SBMJOB (Submit Job) Command V5-154
SBMNETJOB (Submit Network Job) Command V5-160
SBMRJEJOB (Submit RJE Job) Command V5-162
SBMRMTCMD (Submit Remote Command) Command V5-165
SETATNPGM (Set Attention Program) Command V5-169
SETKBDMAP (Set Keyboard Map) Command V5-171
SETPGMINF (Set Program Information) Command V5-174
SIGNOFF (Sign Off) Command V5-177
SLTCMD (Select Command) Command V5-178
SNDBRKMSG (Send Break Message) Command V5-179
SNDDOC (Send Document) Command V5-181
SNDDST (Send Distribution) Command V5-182
SNDEMLIGC (Send Japanese 3270PC Emulation Code) Command V5-189
SNDF (Send File) Command V5-190
SNDJRNE (Send Journal Entry) Command V5-192
SNDMSG (Send Message) Command V5-194
SNDNETF (Send Network File) Command V5-196
SNDNETMSG (Send Network Message) Command V5-198
SNDNETSPLF (Send Network Spool File) Command V5-199
SNDPGMMSG (Send Program Message) Command V5-201
SNDPTFORD (Send Program Temporary Fix Order) Command V5-207
SNDRCVF (Send/Receive File) Command V5-208
SNDRPY (Send Reply) Command V5-210
SNDSRVRQS (Send Service Request) Command V5-211
SNDTIEF (Send Technical Information Exchange File) Command V5-212
SNDUSRMSG (Send User Message) Command V5-213
STRAPF (Start Advanced Printer Function) Command V5-217

STRAPW (Start Advanced Print Writer) Command V5-218
STRBAS (Start Basic) Command V5-219
STRBASPRC (Start Basic Procedure) Command V5-220
STRBGU (Start Business Graphics Utility) Command V5-221
STRCBLDBG (Start COBOL Debug) Command V5-222
STRCGU (Start CGU) Command V5-223
STRCMTCTL (Start Commitment Control) Command V5-224
STRCPYSCN (Start Copy Screen) Command V5-226
STRDBG (Start Debug) Command V5-228
STRDBRDR (Start Data Base Reader) Command V5-230
STRDFU (Start DFU) Command V5-232
STRDKTRDR (Start Diskette Reader) Command V5-234
STRDKTWTR (Start Diskette Writer) Command V5-236
STRDSKCOL (Start Disk Data Collection) Command V5-238
STREDU (Start Education) Command V5-239
STREML3270 (Start 3270 Display Emulation) Command V5-240
STRFMA (Start Font Management Aid) Command V5-244
STRIDD (Start Interactive Data Definition Utility) Command V5-245
STRIDXSCH (Start Index Search) Command V5-246
STRITF (Start Interactive Terminal Facility) Command V5-247
STRJOBTRC (Start Job Trace) Command V5-248
STRJRNAP (Start Journal Access Path) Command V5-250
STRJRNPF (Start Journal Physical File) Command V5-252
STRKPF (Start Kanji Printer Function) Command V5-254
STRMOD (Start Mode) Command V5-255
STROFC (Start Office) Command V5-256
STRPASTHR (Start Pass-Through) Command V5-257
STRPCCMD (Start PC Command) Command V5-260
STRPCO (Start PC Support Organizer) Command V5-261
STRPDM (Start Programming Development Manager) Command V5-262
STRPFRMON (Start Performance Monitor) Command V5-263
STRPFRT (Start Performance Tools) Command V5-266
STRPGMMNU (Start Programmer Menu) Command V5-267
STRPPW (Start Page Printer Writer) Command V5-271
STRPRTEML (Start Printer Emulation) Command V5-272
STRPRTWTR (Start Printer Writer) Command V5-276
STRQRY (Start Query) Command V5-279
STRQST (Start Question and Answer) Command V5-280
STRRJECSL (Start RJE Console) Command V5-281
STRRJERDR (Start RJE Reader) Command V5-282
STRRJESSN (Start RJE Session) Command V5-283
STRRJEWTR (Start RJE Writer) Command V5-285
STRSAM (Start Sampled Address Monitor) Command V5-289
STRSAMCOL (Start Sampled Address Monitor Data Collection) Command . . V5-291
STRSBS (Start Subsystem) Command V5-292
STRSDA (Start SDA) Command V5-294
STRSEU (Start SEU) Command V5-296
STRSPTN (Start Support Network) Command V5-298
STRSQL (Start Structured Query Language) Command V5-299
STRSRVJOB (Start Service Job) Command V5-301
STRSST (Start System Service Tools) Command V5-302
STRS36 (Start System/36) Command V5-303
STRS36PRC (Start System/36 Procedure) Command V5-304
STRTIESSN (Start Technical Information Exchange Session) Command V5-305
STRWP (Start Word Processing) Command V5-306
TFRBCHJOB (Transfer Batch Job) Command V5-307
TFRCTL (Transfer Control) Command V5-309
TFRGRPJOB (Transfer to Group Job) Command V5-311
TFRJOB (Transfer Job) Command V5-313
TFRSECJOB (Transfer Secondary Job) Command V5-315
TRCINT (Trace Internal) Command V5-316
TRCJOB (Trace Job) Command V5-319
UPDDTA (Update Data) Command V5-322

VFYCMN (Verify Communications) Command V5-323
VFYPRT (Verify Printer) Command V5-324
VRYCFG (Vary Configuration) Command V5-325
WAIT (Wait) Command V5-328
WRKACTJOB (Work with Active Jobs) Command V5-330
WRKALR (Work with Alerts) Command V5-332
WRKAUTL (Work with Authorization Lists) Command V5-334
WRKCFGL (Work with Configuration Lists) Command V5-335
WRKCFGSTS (Work with Configuration Status) Command V5-336
WRKCMD (Work with Commands) Command V5-338
WRKCNTINF (Work with Contact Information) Command V5-339
WRKCOSD (Work with Class-of-Service Descriptions) Command V5-340
WRKCTLD (Work with Controller Descriptions) Command V5-341
WRKDBFIDD (Work with DB Files using IDDU) Command V5-342
WRKDDMF (Work with Distributed Data Management File) Command V5-343
WRKDEVD (Work with Device Descriptions) Command V5-345
WRKDEVTBL (Work with Device Table) Command V5-347
WRKDIR (Work with Directory) Command V5-348
WRKDOC (Work with Documents) Command V5-350
WRKDOCLIB (Work with Document Libraries) Command V5-351
WRKDOCPRTQ (Work with Document Print Queue) Command V5-352
WRKDPCQ (Work with DSNX/PC Distribution Queues) Command V5-353
| WRKDSKSTS (Work with Disk Status) Command V5-354
WRKDSTL (Work with Distribution List) Command V5-355
WRKDSTQ (Work with Distribution Queue) Command V5-357
WRKDTADCT (Work with Data Dictionaries) Command V5-358
WRKDTADFN (Work with Data Definitions) Command V5-359
WRKF (Work with File) Command V5-360
WRKFCT (Work with Forms Control Table) Command V5-362
WRKFLR (Work with Folders) Command V5-363
WRKHDWPRD (Work with Hardware Products) Command V5-364
WRKJOB (Work with Job) Command V5-365
WRKJOBD (Work with Job Description) Command V5-367
WRKJOBQ (Work with Job Queue) Command V5-368
WRKJRN (Work with Journal) Command V5-369
WRKJRNA (Work with Journal Attributes) Command V5-370
WRKLIB (Work with Library) Command V5-371
WRKLIBPDM (Work with Libraries Using PDM) Command V5-372
WRKLIND (Work with Line Descriptions) Command V5-373
WRKMBRPDM (Work with Members Using PDM) Command V5-374
WRKMNU (Work with Menus) Command V5-376
WRKMODD (Work with Mode Descriptions) Command V5-377
WRKMSGD (Work with Message Descriptions) Command V5-378
WRKMSGQ (Work with Message Queues) Command V5-379
WRKNETF (Work with Network Files) Command V5-380
WRKNETJOBE (Work with Network Job Entry) Command V5-382
WRKOBJ (Work with Object) Command V5-383
| WRKOBJOWN (Work with Objects by Owner) Command V5-385
WRKOBJLCK (Work with Object Lock) Command V5-386
WRKOBJPDM (Work with Objects Using PDM) Command V5-388
WRKOUTQ (Work with Output Queue) Command V5-390
WRKOUTQD (Work with Output Queue Description) Command V5-391
WRKPGMTBL (Work with Program Table) Command V5-392
WRKPRB (Work with Problem) Command V5-393
WRKPRDINF (Work with Product Information) Command V5-394
WRKQRY (Work with Query) Command V5-395
WRKQST (Work with Question) Command V5-396
WRKRDR (Work with Readers) Command V5-397
WRKRJESSN (Work with RJE Session) Command V5-398
WRKRPYLE (Work with System Reply List Entry) Command V5-399
WRKSBMJOB (Work with Submitted Jobs) Command V5-400
WRKSBS (Work with Subsystems) Command V5-401
WRKSBSD (Work with Subsystem Description) Command V5-402

WRKSBSJOB (Work with Subsystem Jobs) Command V5-403
WRKSOC (Work with Sphere of Control) Command V5-405
WRKSPLF (Work with Spooled Files) Command V5-406
WRKSPLFA (Work with Spooled File Attributes) Command V5-408
WRKSSND (Work with Session Description) Command V5-409
WRKSYSSTS (Work with System Status) Command V5-410
WRKTIE (Work with Technical Information Exchange) Command V5-411
WRKTXTPRF (Work with Text Profiles) Command V5-412
WRKUSRJOB (Work with User Jobs) Command V5-413
WRKUSRPRF (Work with User Profiles) Command V5-414
WRKUSRTBL (Work with User Table) Command V5-415
WRKWTR (Work with Writers) Command V5-416

Command Descriptions RNMM through WRKWTR

Volumes 2 through 5 describe every control language command, including commands for the Operating System/400[1] (OS/400[1]) and commands for languages and utilities, and show their syntax.

Volume 2 contains commands ADDACC through CPYDOC.

Volume 3 contains commands CPYF through DLTPGM.

Volume 4 contains commands DLTPNLGRP through RNMDLO.

Volume 5 contains commands RNMM through WRKWTR.

RNMM (Rename Member) Command

The Rename Member (RNMM) command changes the name of a specified file member. The member cannot be renamed while it is in use; other users can read but not change other members in the file that contains the member being renamed. A member that is opened in the same job can be renamed.

Restriction: To rename a file member, you must have object management and object operational authority for the file that contains the member. You also must have read and update authority to the file's library.

```
                                                                   Required
             ┌►*LIBL/────────┐
RNMM──FILE───┼─*CURLIB/──────┼──data-base-file-name──────────────────────►
             └─library-name/─┘

                                                 ┌─┐
►──MBR member-name───NEWMBR new-member-name──────┤P├──────────────────────
                                                 
                                                  Job: B,I  Pgm: B,I  Exec
```

FILE Parameter
Specifies the qualified name of the data base file (physical or logical) that contains the member to be renamed (library-name/file-name).

The possible library values are:

***LIBL:** The library list is used to locate the file.

***CURLIB:** The current library for the job is used to locate the file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the file is located.

MBR Parameter
Specifies the name of the physical or logical file member that is renamed.

NEWMBR Parameter
Specifies the new name of the member being renamed. The member remains in the same file. The new name must be unique in the file.

---

[1] Operating System/400 and OS/400 are trademarks of the International Business Machines Corporation.

Example

```
RNMM   FILE(ELEMENT)  MBR(LEAD)  NEWMBR(GOLD)
```

This command renames member LEAD in file ELEMENT as GOLD. The library list (*LIBL) is used to find the file.

RNMOBJ (Rename Object) Command

The Rename Object (RNMOBJ) command changes the name of an object in a library. The new name specified for the object must be unique in the library for the object type. If the object being renamed is in use when the command is entered, the command is not run and a message is sent to the user who enters the command. If a library is on an active user's library list when the library is renamed, a Display Library List (DSPLIBL) command reflects the new name. (Renaming a library can cause programming errors, and therefore is not recommended.)

Restrictions: (1) The user must have object operational authority for the object that is being renamed; he must also have update rights for the library in which the object is located. (2) User profiles, edit descriptions, line descriptions, device descriptions, control unit descriptions, and journals and journal receivers cannot be renamed. (3) The following objects cannot be renamed: the job's temporary library QTEMP, the system library QSYS, the system operator message queue QSYSOPR, all work station user message queues, and the system log QHST. In addition, a PL/I program cannot be renamed after it has been created.

```
                 ┌►*LIBL/───────┐
RNMOBJ───OBJ─────┼─*CURLIB/─────┼──object-name──────────────────────────►
                 └─library-name/┘

              ┌──────────────────────────────────┐
              │ Select one of the following:     │
              │ *AUTL      *JOBD      *PRDDFN    │
              │ *CHTFMT    *JOBQ      *QRYDFN    │
►───OBJTYPE───│ *CLS       *LIB       *RCT       │───NEWOBJ new-object-name────►
              │ *CMD       *MENU      *SBSD      │
              │ *DTAARA    *MSGF      *SCHIDX    │
              │ *DTAQ      *MSGQ      *SPADCT    │
              │ *FCT       *OUTQ      *SSND      │
              │ *FILE      *PGM       *TBL       │
              │ *GSS       *PNLGRP               │
              └──────────────────────────────────┘
                                                                   Required
───────────────────────────────────────────────────────────────────────────
                                                                   Optional
                                ┌─┐
                ┌►*LCL──────┐   │P│
►───SYSTEM──────┼─*RMT──────┼───┴─┴──────────────────────────────────────
                └─*FILETYPE─┘

                                               ┌──────────────────────────┐
                                               │ Job: B,I  Pgm: B,I  Exec │
                                               └──────────────────────────┘
```

OBJ Parameter

Specifies the current qualified name of the object that is renamed. (If no library qualifier is given, *LIBL is used to find the object.) The object name should be qualified to ensure that the right object is renamed.

The possible library values are:

<u>*LIBL</u>: The library list is used to locate the object.

***CURLIB:** The current library for the job is used to locate the object. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the object is located.

OBJTYPE Parameter

Specifies the type of the object that is renamed. Enter one of the following OS/400 object types:

Value	**Type of OS/400 Object**
*AUTL	Authorization List
*CHTFMT	Chart format
*CLS	Class
*CMD	Command
*DTAARA	Data area
*DTAQ	Data queue
*FCT	Forms control table
*FILE	File
*GSS	Graphics symbol set
*JOBD	Job description
*JOBQ	Job queue
*LIB	Library

*MENU	Menu
*MSGF	Message file
*MSGQ	Message queue
*OUTQ	Output queue
*PGM	Program
*PNLGRP	Panel Group
*PRDDFN	Product definition
*QRYDFN	Query Definition
*RCT	Reference code translate table
*SBSD	Subsystem description
*SCHIDX	Search Index
*SPADCT	Spelling aid dictionary
*SSND	Session description
*TBL	Table

NEWOBJ Parameter

Specifies the new name of the object being renamed. The object remains in the same library. Enter the new name that identifies the object to the system.

SYSTEM Parameter

Specifies if the information returned by RNMOBJ is about files on the local system (*LCL), about files on the remote system or systems (*RMT), or about files on both local and remote systems (*ALL). For performance considerations, unless explicitly looking for information about files on systems other than the local system, the default value of *LCL should not be changed.

*LCL: There is no change in the function of this command.

***RMT:** The remote file referred to by the source DDM file is renamed.

Note: If the user wants to rename a remote file, two DDM files must be used. The existing DDM file goes in the OBJ value and the new DDM file goes in the NEWOBJ value. The new DDM file must be in the same library as the existing DDM file. When the remote rename occurs, it is the the remote file name in the existing DDM file that is renamed to the new DDM file name.

***FILETYPE:** If the OBJ name is a DDM file, then the NEWOBJ name is a DDM file, but if the OBJ name is not a DDM file, the renaming is local.

Example

```
RNMOBJ  OBJ(PAYROLL/FILEX)  OBJTYPE(*FILE)  NEWOBJ(MSTR)
```

The library named PAYROLL is searched for the file named FILEX. If the file is found and if the user has object operational authority for FILEX and update authority for the PAYROLL library, FILEX is renamed MSTR.

ROLLBACK (Rollback) Command

The Rollback (ROLLBACK) command is used to start the current transaction over and reestablish the last commitment boundary as the current commitment boundary.

Any changes made to the files under commitment control since the last commitment boundary was established are rolled back (not made permanent in the files). Any records locked by commitment control are released.

More information on the use of commitment control is in the *Backup and Recovery Guide*.

```
                                                                   Optional
ROLLBACK──────────────────────────────────────────────────────────────────

                                                     Job: B,I  Pgm: B,I  Exec
```

There are no parameters for this command.

Example

```
ROLLBACK
```

This command reestablishes the last commitment boundary (the point at which a COMMIT command was last issued). Any updates, additions, or deletions made to the file's data since that commitment boundary are rolled back or removed, and the original entries are put back in the files. Any records that were added to the files remain as deleted records. The file is repositioned to the last commitment boundary. All record locks for files under commitment control are released.

RPLDOC (Replace Document) Command

The Replace Document (RPLDOC) command replaces information in a document.

Restriction: (1) If the current user of this command is working on behalf of another user, he must have the authority granted to him by the other user with the Grant User Permission (GRTUSRPMN) command to work on his behalf. (2) The document must be checked out before using the Retrieve Document (RTVDOC) command before using this command.

```
                ┌document-name┐
RPLDOC──TODOC───┤             ├──────────────────────────────────────────────►
                └*DOCID───────┘

                                                   ┌─┐
           ┌►*LIBL/────────┐                       │P│
►──DOCFILE─┼─*CURLIB/──────┼──data-base-file-name──┴─┴───────────────────────►
           └─*library-name/┘
                                                                      Required
──────────────────────────────────────────────────────────────────────────────
                                                                      Optional
         ┌►*NONE──────┐         ┌►*NONE───────────────────────────────┐
►──TOFLR─┤            ├──DOCID─┤                                     ├──────►
         └folder-name─┘         └'library-assigned-document-name'─────┘

           ┌►*FIRST─────┐            ┌►*DFT─────────────────┐
►──DOCMBR──┤            ├──DOCTYPE──┼─*FFT─────────────────┼────────────────►
           └member-name─┘            ├─*RFT─────────────────┤
                                     └─document-type-number─┘

         ┌►*CURRENT──────────┐
►──USRID─┤                   ├───────────────────────────────────────────────►
         └user-ID address────┘

              ┌►*SYSVAL─────────────────────────────┐
►──DOCCHRID───┼─*DEVD───────────────────────────────┼────────────────────────►
              └─graphic-character-set  code-page────┘

      (1)     ┌►*SYSVAL─────────────────────────────┐
►──CMDCHRID───┼─*DEVD───────────────────────────────┼────────────────────────
              └─graphic-character-set  code-page────┘

(1) This parameter value applies to the USRID parameter.

                                              ┌─────────────────────────┐
                                              │Job: B,I  Pgm: B,I  Exec │
                                              └─────────────────────────┘
```

TODOC Parameter

Specifies the name of the document where the data is being placed, or the DOCID parameter is used to specify the library-assigned document name where the data is being replaced.

document-name: Specify the user-assigned name of the document to replaced.

DOCID: The document being replaced is identified by the library-assigned document name specified on the DOCID parameter.

DOCFILE Parameter

Specifies the name of the data base file that contains the document data being filed. The data base file can be user-defined or, the outfile from the Receive Distribution (RCVDST) or the Retrieve Document (RTVDOC) commands. Data base files that are used to replace document data are used as is with any changes. If an outfile is specified, only the document data record format is read from the outfile and the prefix is removed from the document data. More information on defining the format of data base files (OUTFILES) is in the *Data Base Guide.*

The possible library values are:

***LIBL:** The library list is used to locate the data base file.

***CURLIB:** The current library for the job is used to locate the data base file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the data base file is located.

TOFLR Parameter

Specifies the name of the folder that contains the document being replaced.

***NONE:** No folder is specified when the document is identified by the DOCID parameter.

folder-name: Specify the name of the folder that contains the document being replaced. A folder name can consist of a series of folder names if the document being replaced is located in a folder that is contained in another folder. A maximum of 63 characters can be specified.

DOCID Parameter

Specifies the library-assigned name of the document replaced. This is the name assigned to the document by the system when it is created. The library-assigned document names can be determined by using the Query Document Library (QRYDOCLIB) command or the library-assigned name is returned in a completion message when using the File Document (FILDOC) command.

Library-assigned document names are 24 characters in length with the following format: YYYYMMDDHHMNSSHSSNSNSNSN where :

YYYY = year
MM = month
DD = day
HH = hour
MN = minute
SS = second
HS = hundredths of a second
SNSNSNSN = system name

***NONE:** No library-assigned document name is required when the document is identified by the TODOC parameter.

'library-assigned-document-name': Specify the library-assigned document name to be replaced. (More information on library-assigned document names is in the *Communications: Distribution Services Network Administrator's Guide*.)

Note: When the user is replacing a document and a cancel request is enter, the document may or may not be replaced.

DOCMBR Parameter

Specifies the document data base file member that contains the data to replace the current data in the document.

***FIRST:** The first member (in the order it was created) in the data base file contains the replacement data.

member-name: Specify the name of the data base file that contains the replacement data.

DOCTYPE Parameter

Specifies the type of replacement document. This identifier is used by the system to determine if it can or cannot handle the data stream correctly.

The system code (SYSCOD on the File Document (FILDOC) command) cannot be changed when the document contents are replaced.

***DFT:** The system creates the appropriate document type identifier depending on where the data comes from.

Document type	*Description*
2	Final form text document
3	5520 revisable form text document
4	Word processing EBCDIC
5	Word processing information file
6	Image-data subset document
7	3730 text data stream
8	DIA document library descriptor document
9	3732 display document data stream
10	DIA display document data stream
11	Revisable form text document
12	1403 compatible data stream with variable length, unblocked records
13	Digitized ADS audio
14	IBM PC data file
15	Hard copy of document
223	Data file

If the document file is an OUTFILE created by the RCVDST or RTVDOC command, then the document type is taken from this file. If the file does not contain a document type or if the value in this file is document type 0, then an error message is sent. If the document file is a user-defined file, then the document type is 223 (data file).

***FFT:** The document is Final Form Text. This type of document is intended to be shown and printed and not edited by the receiver.

***RFT:** The document is Revisable Form Text. This type of document can be shown, printed, and edited by the receiver.

document-type-number: Specify a number from 2 through 65,535. The numbers from 2 through 32,767 are controlled by registering them with the IBM Document Interchange Architecture and are used for IBM-defined document types. The numbers 32,768 through 65,535 are not registered with IBM and can be used for non-IBM defined document types.

USRID Parameter

Specifies the name of the user ID and address associated with this request.

***CURRENT:** Replacement is done for the current user of this command.

user-ID/address: Specify the user id address of the user whose document is replaced. The current user of this command must have the authority to work on behalf of the named user ID. Authority to work on behalf of another user is given with the Grant User Permission (GRTUSRPMN) command. If a user files a document on behalf of another user's user-ID, the other user is the owner of the document.

DOCCHRID Parameter

Specifies the character identifier (graphic character set and code page) for the data being filed. The character identifier is related to the display station that was used to create the data being filed. (More information about CHRID processing is in the *Data Base Guide*.)

***SYSVAL:** The system determines the graphic character set and code page values for the command parameter from QCHRID system value.

***DEVD:** The system determines the graphic character set and code page values for the command parameter from the display device description where this command was specified. This option is valid only when specified from an interactive job. If specified in an interactive CL program or a batch job, an error occurs.

graphic-character-set code-page: Specify the graphic character set and values that are used to create the data being filed.

CMDCHRID Parameter

Specifies the character identifier (graphic character set and code page) for the data being entered as command parameter values (applies to USRID parameter). The character identifier is related to the display device that was used to enter the command. (More information about CHRID processing is in the *Data Base Guide*.)

This value translates the USRID parameter to character set and code page of '930 256'. The *Communications: Distribution Services Network Administrator's Guide* contains the character set and code page table for '930 256'.

***SYSVAL:** The system is to determine the graphic character set and code page values for the command parameter from QCHRID system value.

***DEVD:** The system is to determine the graphic character set and code page values for the command parameter from the display device description where this command was entered. This option is valid only when entered from an interactive job. If specified in an interactive CL program or a batch job, an error will occur.

graphic-character-set code-page: Specify the graphic character set and values that were used to create the command parameter. Both parts can be up to three characters in length and must be separated by at least one blank.

Example

```
RPLDOC  TODOC(*DOCID)  DOCFILE(*LIBL/MYFILE)  +
  DOCID('1987060710102053SYSTEM1')  DOCTYPE(*FFT)
```

This command replaces the document with data in the file MYFILE. The data is placed in the document identified by the document identifier '1987060710102053SYSTEM1'. The document type is changed to Final Form Text.

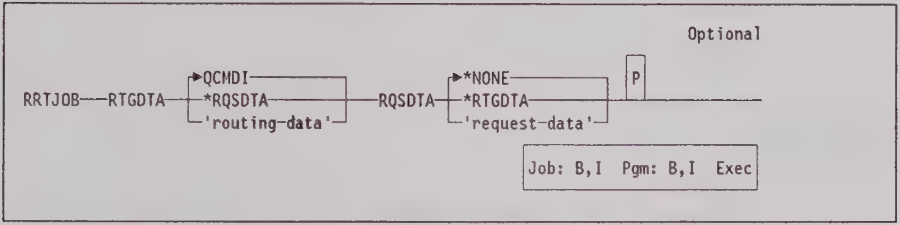
RRTJOB (Reroute Job) Command

The Reroute Job (RRTJOB) command causes a new routing step to start for a job in the current subsystem. For example, the job may need rerouting so that it runs under a different class or in a different storage pool. The rerouting requires changes in the routing data for the job; the new routing data calls a different program used with the new routing step.

When this command is used, any objects allocated in the previous routing step are deallocated and any open files are closed. If the objects or files are needed in the new routing step, they must be allocated or opened again.

Note: Running of this command in a batch job causes loss of spooled inline files because they cannot be accessed in the new routing step. Also, if the RRTJOB command is run while the system is ending (by running of a End System (ENDSBS) command, End System (ENDSYS) command, or the Power Down System (PWRDWNSYS) command), a new routing step is not started and the job is ended.

Restriction: The job must not be a group job.



RTGDTA Parameter

Specifies the routing data used with this job description to start jobs. The routing data is used to determine the routing entry (in the subsystem description) that identifies the program where the job runs.

QCMDI: This routing data matches a routing entry in the IBM-supplied subsystem description (QINTER), which indicates a routing step that is processed by the IBM-supplied control language processor, QCMD, in the QSYS library.

*RQSDTA: The first 80 characters of the request data specified in the RQSDTA parameter of this command is also used as the routing data for the next routing step.

'routing-data': Specify the character string used as the routing data for starting the next routing step. A maximum of 80 characters can be entered (enclosed in apostrophes if necessary).

RQSDTA Parameter

Specifies the request data added on the end of the job's message queue for use by the new routing step. For example, if RTGDTA (QCMDB) is specified, the IBM-supplied batch subsystem QBATCH is being used, and a CL command is supplied here, it becomes a message that is read by the control language processor, QCMD (if the submitted job is routed to QCMD).

*NONE: Specifies that no request data is placed in the job's message queue.

*RTGDTA: Specifies the routing data specified in the RTGDTA parameter is also placed at the end of the job's message queue.

'request-data': Specify the character string placed at the end of the job's message queue for use by the new routing step or some subsequent routing step in the job. A maximum of 256 characters can be entered (enclosed in apostrophes if necessary). When a CL command is entered, it must be enclosed in single apostrophes, and where apostrophes would normally be used *inside* the command, double apostrophes must be used instead.

Example

```
RRTJOB  RTGDTA(INQUIRY)
```

This command reroutes the job in which the command is issued by starting a new routing step with the routing data INQUIRY. The job remains in the same subsystem.

RSMBKP (Resume Breakpoint) Command

The Resume Breakpoint (RSMBKP) command causes a program to continue processing after it has been stopped at a breakpoint. The program that continues is the one that most recently stopped at a breakpoint. When more than one program in the job is stopped at a breakpoint, the End Request (ENDRQS) command can be used to return to the command entry display for a previous program call that is also stopped at a breakpoint.

Restriction: This command is valid only in debug mode and only when the program is stopped at a user-defined breakpoint. That is, this command is not valid at a breakpoint caused by an unmonitored message. To start debug mode, refer to the STRDBG (Start Debug) command.

```
RSMBKP────────────────────────────────────────────────────────────────

                                                   ┌──────────────────────┐
                                                   │Job: I  Pgm: I  Exec  │
                                                   └──────────────────────┘
```

There are no parameters for this command.

Example

`RSMBKP`

Assuming that the program having control is stopped at a breakpoint, this command causes the program to continue, starting from the breakpoint location.

RSMCTLRCY (Resume Controller Recovery) Command

The Resume Controller Recovery (RSMCTLRCY) command resumes error recovery procedures for a specific controller. The error recovery procedures can be ended by the End Controller Recovery (ENDCTLRCY) command or by responding to a failure-related inquiry message with a cancel option.

The RSMCTLRCY command allows you to resume automatic error recovery procedures after they have been stopped, and to reactivate a controller after it has been canceled (if you entered the C response to the inquiry message associated with a controller failure). When the controller is canceled with the C response, all jobs are ended; once the controller is repaired and the RSMCTLRCY command is entered, jobs are allowed to start using the controller again.

Restriction: To use this command, you must have object operational rights for the controller.

```
                                                        Required
                                 ┌─┐
                                 │P│
RSMCTLRCY───CTL controller─name──┴─┴─────────────────────────────
                                              ┌─────────────────────────┐
                                              │Job: B,I  Pgm: B,I  Exec │
                                              └─────────────────────────┘
```

CTL Parameter

Specifies the controller whose recovery is started again. Valid types of controllers are:

CTL Value	Controller
5251	Display station
*PU2	Physical unit (type 2); SDLCs for basic BSC and RJE
*BSC	BSC device (basic BSC and RJE)
*BSCT	BSC device (Multipoint tributary and 3270 Device Emulation)
*APPC	Advanced program-to-program communications
*WSC	Local work station
*WSCE	Local work station (extended)

Specify the name of the controller (as specified in the controller description) for which error recovery procedures are to resume.

Example

```
RSMCTLRCY  CTL(TROLL3)
```

This command resumes error recovery procedures for the controller TROLL3.

RSMDEVRCY (Resume Device Recovery) Command

The Resume Device Recovery (RSMDEVRCY) command resumes error recovery procedures for a specific device. The error recovery procedures are ended by the ENDDEVRCY command or by responding to a failure-related inquiry message with a cancel option.

The RSMDEVRCY command allows you to resume automatic error recovery procedures after they have been stopped, and to reactivate a device after it has been canceled (if you entered the C response to the inquiry message associated with a device failure). When the device is canceled with the C response, all jobs are ended; once the device is repaired and the RSMDEVRCY command is entered, jobs are allowed to start using the device again.

Restriction: To use this command, you must have object operational rights for the device.

```
                                                        Required
                         ┌─┐
                         │P│
RSMDEVRCY──DEV device-name─┴─┴───────────────────────────────────────
                                               ┌─────────────────────┐
                                               │Job: B,I  Pgm: B,I  Exec│
                                               └─────────────────────┘
```

DEV Parameter
Specifies the device whose recovery is started again. Valid types of devices are:

DEV Value	Device
5219	Printer (work station)
5224	Printer (work station)
5225	Printer (work station)
5251	Display station
5252	Dual display station
5256	Printer (work station)
5291	Display station
5292	Display station
*PLU1	Physical unit (type 1)
*BSC	BSC device
*BSCT	BSC multipoint tributary
*APPC	Advanced program-to-program communications

Specify the name of the device (as specified in the device description) for which error recovery procedures are to resume.

Example

```
RSMDEVRCY  DEV(WSPR03)
```

This command resumes error recovery procedures for the device WSPR03 to resume.

RSMLINRCY (Resume Line Recovery) Command

The Resume Line Recovery (RSMLINRCY) command resumes error recovery procedures for a specific line. The error recovery procedures are ended by the End Line Recovery (ENDLINRCY) command or by responding to a failure-related inquiry message with a cancel option.

The RSMLINRCY command allows you to resume automatic error recovery procedures after they stop, and to reactivate a line after it is canceled (if you enter the C response to the inquiry message associated with a line failure). When the line is canceled with the C response, all jobs are ended; once the line is repaired and the RSMLINRCY command is entered, jobs are allowed to start using the line again.

Restriction: To use this command, you must have object operational rights for the line.

```
                                                  Required
                             ┌─┐
                             │P│
RSMLINRCY──LINE line-name────┴─┴──────────────────────────────
                                       ┌────────────────────────┐
                                       │Job: B,I  Pgm: B,I  Exec│
                                       └────────────────────────┘
```

LINE Parameter
Specifies the line whose recovery is started again.

Example

```
RSMLINRCY  LINE(NYC2)
```

This command resumes error recovery procedures for the line NYC2 to resume.

RSTAUT (Restore Authority) Command

The Restore Authority (RSTAUT) command restores the authority of object use to user profiles. This command restores the same object use authority to specified objects in the user profile that each user profile had when all the profiles were saved by the Save System (SAVSYS) command. It allows existing authorities, given after the save, to remain. Authority is not restored to the user profiles until the profiles are first restored to the system by the Restore User Profiles (RSTUSRPRF) command and all the objects (for which authority is being given) are restored to the same libraries by the Restore Library (RSTLIB) or Restore Object (RSTOBJ) commands where they were saved.

If the whole system is being restored, the following sequence must be followed. The RSTAUT command must be the last step in the sequence.

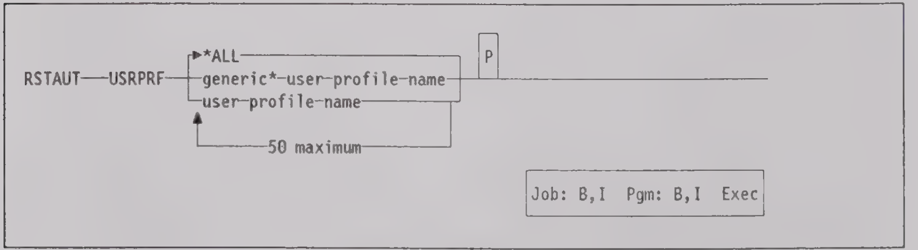
1. Install the operating system (alternative method to load the program): Restores the QSYS library and ensures that the IBM-supplied user profiles are there.
2. RSTUSRPRF command: Restores all the saved user profiles to the system. (*ALL is the default for the USRPRF parameter.)
3. RSTLIB command: Restores all the user libraries (including the other IBM-supplied libraries). To restore them in one operation, the SAVLIB parameter must specify *NONSYS.
4. RSTAUT command: Restores the object use authority to user profiles.

Note: Steps 2 through 4 can be done more than once. For example, after the user profiles are restored (step 2), you can restore only critical application libraries (step 3), followed by a restore of object authority (step 4). This supplies an operational system limited to using only the critical libraries. Later, the remaining user libraries can be restored, followed by the operations to restore user profiles and object authority.

If one user profile is being restored, the following sequence must be followed. The RSTAUT command must be the last step.

1. RSTUSRPRF command: Restores the specified user profiles to the system.
2. RSTLIB command: Restores the specified user libraries to the system. If a user profile is being restored because the current profile on the system is damaged, then the needed libraries already exist on the system and restoring of the libraries is not necessary.
3. RSTAUT command: Restores the object use authority to user profiles. The specified profiles may include profiles that were restored using multiple RSTUSRPRF commands.

Restrictions: (1) To use this command, you must have the save system special authority in your user profile, specified by SPCAUT(*SAVSYS). (2) All subsystems must be ended (ENDSYS or ENDSBS(*ALL)) before running this command.



USRPRF Parameter
Specifies the names of one or more user profiles to have their private authorities restored. The specified user profiles must first be restored using the RSTUSRPRF command.

*ALL: Specifies all of the user profiles that are restored but do not have their private authorities restored. This includes user profiles that were restored using multiple RSTUSRPRF commands.

generic-user-profile-name:* Specify one or more generic names of sets of user profiles. (Add an asterisk after the last character in a generic name; for example, ABC*. If an * is not included with the name, the system assumes that the name is a complete user profile name.)

user-profile-name: Specify one or more names of specific user profiles. Both generic names and specific names can be specified in the same command.

Examples

```
RSTAUT
```

This command restores to each user profile the authority to use each object that the profile had at the time when the system was saved. The user profiles and the libraries and their objects must be restored before the RSTAUT command is sent.

```
RSTUSRPRF USRPRF(USER1 USER2 USER3 USER4)

RSTLIB LIB(USERLIB)

RSTAUT USRPRF(USER1 USER2 USER3)
```

To each specified user profile that was successfully restored, the RSTAUT command restores the authority to use each object that the profile had at the time the system was saved. The user profile(s) and the libraries and their objects must be restored before the RSTAUT command is sent. Because USER4 was not specified in the RSTAUT command, its authorities are still available and may be restored at a later date.

```
RSTUSRPRF USRPRF(*ALL)
RSTLIB LIB(USERLIBA)

RSTUSRPRF USRPRF(*ALL)
RSTLIB LIB(USERLIBB)

RSTUSRPRF USRPRF(*ALL)
RSTLIB LIB(USERLIBC)

RSTAUT USRPRF(*ALL)
```

The RSTAUT command restored private authorities for all restored user profiles on the system. This includes authorities for all user profiles restored by the three RSTUSRPRF commands. Other user profiles on the system that did not have their authorities restored before these commands are also restored by the RSTAUT(*ALL) command.

```
RSTUSRPRF USRPRF(USER1 USER2)
RSTLIB LIB(USERLIBA)

RSTUSRPRF USRPRF(USER1 USER3)
RSTLIB LIB(USERLIBB)

RSTAUT USRPRF(*ALL)
```

The RSTAUT command restored private authorities for USER2 and USER3 and for the most recent version of USER1. Because the user profiles have the same name, the second RSTUSRPRF overlays the first version of USER1. Therefore, only the most recent authorities for USER1 are restored.

RSTCFG (Restore Configuration) Command

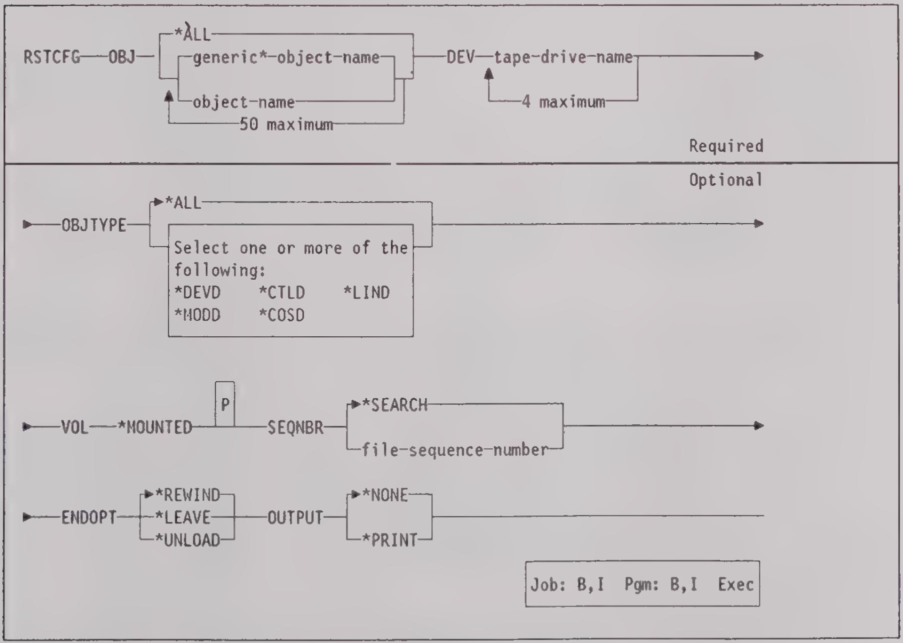
The Restore Configuration (RSTCFG) command restores to the system a device configuration object that was saved to tape by the Save System (SAVSYS) command. The types of objects that can be restored by this command are listed in the OBJTYPE parameter.

The user profile of the security officer (QDFTOWN) becomes the default owner of any objects restored in the system whose owner is not known to the system.

If an object already exists in the library to which it is restored, the public and private authorities of the existing object are kept. If the object does not exist in the library, all public authorities are restored, but any private authorities must be given again.

Restrictions: (1) With the exception of overrides for the restore file, this command ignores all file overrides that are currently in effect for the job. (2) To use this command, you must have either the special authority *SAVSYS specified in your user profile (SPCAUT parameter), or you must have:

1. Read and add authority for, or be the owner of, the specified library.
2. Object existence authority for, or be the owner of, each object specified if the object already exists in the library on the system. If the object does not already exist in the library on the system, you must have add authority for the user profiles that own each object being restored.
3. The device configuration object must be varied off for it to be restored. To vary off a device configuration object, use the Vary Off Configuration (VRYCFG) command.



OBJ Parameter

Specifies the names of one or more objects to restore. Specify the name of each object or the generic name of each group of objects to restore. (Add an asterisk after the last character in the generic name; for example, ABC*. If an * is not included with the name, the system assumes that the name is a complete object name.)

If the OBJTYPE parameter is not specified when the command is specified, all the object types listed in the description of the OBJTYPE parameter are restored, if they are in the specified library on the media and have the specified names.

***ALL:** All the device configuration objects are restored, depending on the values specified for OBJTYPE.

generic-object-name:* Specify one or more generic names of groups of objects in the specified library to restore.

object-name: Specify one or more names of specific objects to restore. (Both generic names and specific names can be specified in the same command.)

DEV Parameter
Specifies the name of the tape device used to restore the objects. The device name must already be known on the system by a device description. If you are using more than one tape drive (up to a maximum of four), specify the names of the drives in the order to be used. If using more than one tape volume, you may want to use more than one tape drive, so that one tape volume can rewind/unload while another tape drive is processing.

OBJTYPE Parameter
Specifies the types of OS/400 objects that are restored.

***ALL:** All of the device configuration object types that are specified by name are restored. If *ALL is also specified for the OBJ parameter, then all of the device configuration objects saved are restored.

object-type: Specify the value for each of the types of objects that are to be restored. The following types can be specified:

*DEVD	Device description
*CTLD	Controller description
*LIND	Line description
*MODD	Mode description
*COSD	Class-of-service description

The object types shown are the ones saved in the device configuration media file by the Save System (SAVSYS) command.

VOL Parameter
Specifies the volume identifiers of the tape volumes from which the objects are restored. The volumes must be put on tape in the same order as they were when the objects were saved. If the objects are known to the system before the restore operation, the system can verify if the volumes put on tape contain the current version of each object. A maximum of 50 entries can be specified. (For an more information on the VOL parameter, refer to Appendix A, "Expanded Parameter Descriptions.")

***MOUNTED:** The objects are restored from the volumes that are currently put on the device specified by the DEV parameter. The version of the objects that is restored is the first version found in the specified location, unless a specific version is identified by the SAVDATE and SAVTIME parameters.

SEQNBR Parameter
Specifies, only when tape is used, which sequence number is used for the restore process.

***SEARCH:** The volume put on a device is searched for a data file with an identifier that is a match for the SAVLIB parameter value; when a match is found, the object is restored. If the last operation on the device specifies ENDOPT(*LEAVE) (the tape is positioned at the location where the last operation ended), the file search starts with the first data file beyond the current tape position. If ENDOPT(*LEAVE) was not used for the last operation (or if the tape was manually rewound after an ENDOPT(*LEAVE) operation), the search starts with the first data file on the volume.

file-sequence-number: Specify the sequence number of the file used for the restore process.

ENDOPT Parameter
Specifies, only when tape is used, what operation is automatically done on the tape volume after the save or restore operation ends. If more than one reel is included, this parameter applies only to the last reel used; all other reels are rewound and unloaded when the end of tape is reached.

***REWIND:** The tape is automatically rewound, but not unloaded, after the save or restore operation has ended.

***LEAVE:** The tape is not rewound; another save or restore operation can start at the current position on the tape.

***UNLOAD:** The tape is automatically rewound and unloaded after the save or restore operation has ended.

OUTPUT Parameter
Specifies if output that shows information about the status of the objects is created. The output shows the parameter values that were specified and shows all objects, restored and not restored. Information relative to each object's security is listed for the restored objects.

***NONE:** No output is created.

***PRINT:** The output is provided with the job's spooled output on a printer.

Examples

```
RSTCFG  DEV(TAP1)  OBJ(*ALL)  OBJTYPE(*ALL)
```

This command restores all of the device configuration objects from the tape on the TAP1 drive.

```
RSTCFG  DEV(TAP1)  OBJ(PRT1)  VOL(ABCD)  OBJTYPE(*DEVD)
```

The device description for PRT1 that was saved on tape volume ABCD is restored back to the system. If device description PRT1 already exists on the system, it must be varied off before it can be restored.

RSTDLO (Restore Document Library Object) Command

The Restore Document Library Object (RSTDLO) command restores documents and folders to the system that were saved on diskette, tape, or a save file.

Restoring a document either replaces the existing document content and control information if the document currently exists on the system or it adds a new document and control information if the document does not currently exist. To find out if the document currently exists, the system searches the document library for another document with the same system object name, document name, folder path name, and owner.

The document and folder name of the document object on the media must be the same as the document name and folder name of the document on the system unless the document is renamed and put in a different folder during the restore.

Note: Folder names must match exactly for restored folders. All objects that are not in use are restored from the folder on the media or in the the save file to the existing folder. Restoring a folder creates a new folder object if the folder currently does not exist and adds to this new folder all objects saved with the folder on the media or in the save file. If the folder already exists, any document or folder objects that do not already exist within it are created as new. The existing documents are replaced with the version from the media. To find out if the folder currently exists, the system searches the document library for another folder with the same system object name, document name, folder path name, and owner.

The creation date of the document is not changed if the document currently exists, and if the document does not currently exist, the creation date is set to the creation date saved with the document when it is created.

This command can be used to restore the documents and folders if the document was or was not freed by the Save Document Library Object (SAVDLO) command, or to restore documents and folders that were deleted by the Delete Document Library Object (DLTDLO) command.

The user profile of the default owner (QDFTOWN) becomes the default owner of any documents and folders restored in the system whose previous owner is not known to the system or is not enrolled in the system distribution directory.

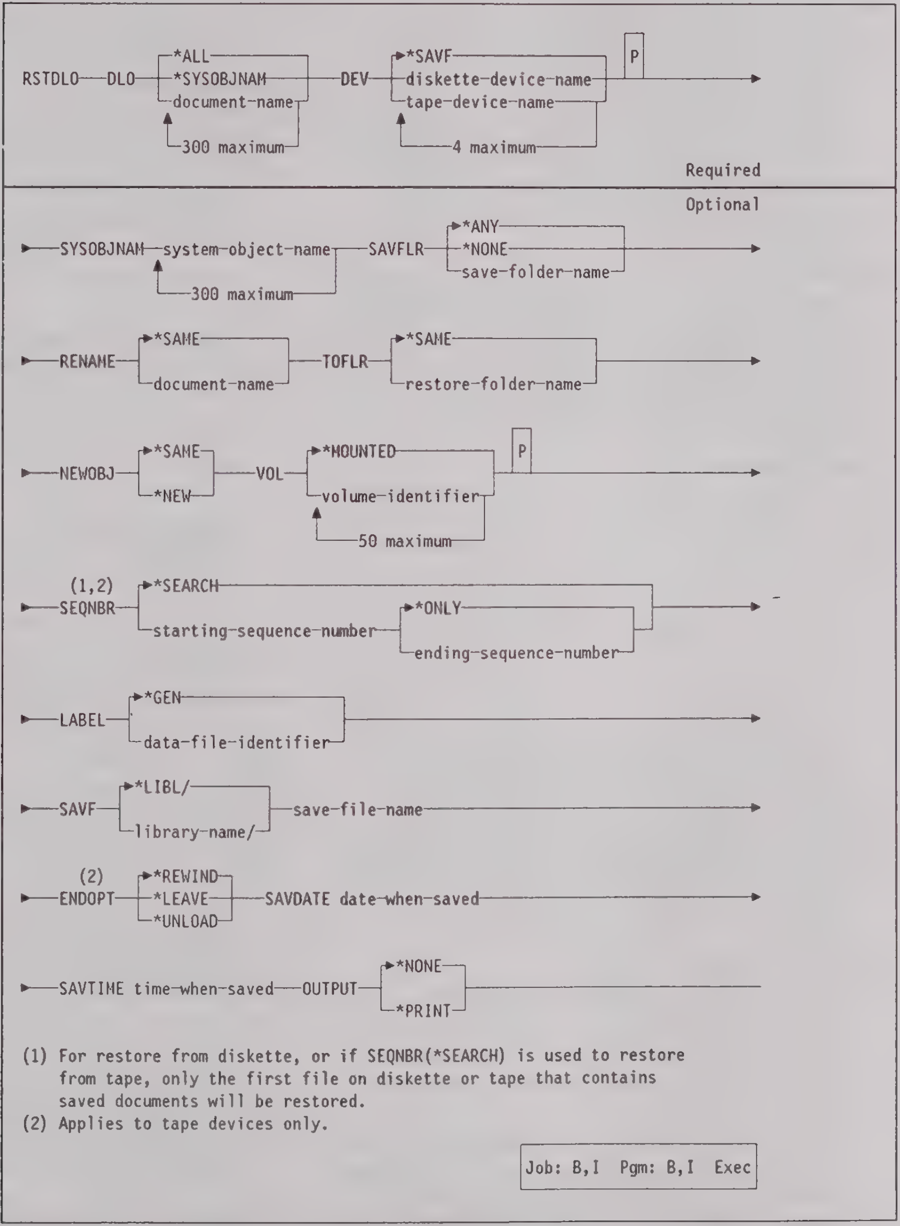
If a document and folder already exist on the system to which it is being restored, the security is not changed.

Note: If the document and folder does not exist, public authority, authorization list, and personal status are restored; however, all other private document and folder authorities are not restored. These authorities must be re-established by the owner.

If this command ends abnormally, objects are left on the system in an unknown state and cannot be found in a library. (This can occur if a power failure occurs when running this command.) The Reclaim Storage (RCLSTG) command can be used to clean up the auxiliary storage and delete those objects from the system.

When a set of documents and folders is restored, all documents and folders in the set must exist in the same save file on the media or in the same online save file. For example, if a document exists in two different media files, only you may control which documents will be restored by specifying the media file using the sequence number (tape) or label (tape or diskette) parameters. If more than one version of the document exists, the SAVDATE and SAVTIME parameters may also be used to select the correct version.

Restrictions: (1) To use this command you must have either *ALLOBJ or *SAVSYS special authority, or *ALL authority to the object being restored if the object currently exists. *CHANGE authority is required for a folder containing documents and folders if the document or folder being restored into it does not currently exist and it to be added to the folder. (2) The restore document support uses journaling and commitment control for the document search files (QUSRSYS,QAOS*). Commitment control should be inactive when this function is requested. If commitment control is active when this function is requested, the journal must be QUSRSYS/QAOSDIAJRN.



(1) For restore from diskette, or if SEQNBR(*SEARCH) is used to restore from tape, only the first file on diskette or tape that contains saved documents will be restored.
(2) Applies to tape devices only.

Job: B,I Pgm: B,I Exec

DLO Parameter

Specifies the names of one or more documents and/or folders that are restored.

***ALL:** All documents and/or folders that are saved on the media and meet the values specified by the SAVFLR parameter are restored.

***SYSOBJNAM:** The documents with the system object names specified in the SYSOBJNAM parameter are restored.

document-name: Specify the user-assigned names of the documents restored. A maximum of 12 characters can be specified for the name and a maximum of 300 document names can be specified.

DEV Parameter

Specifies the name of the device that is used to restore the documents and folders. The device name must already be known on the system by a device description.

***SAVF:** The documents and folders are restored from a save file.

diskette-device-name: Specify the name of the diskette device whose diskette files are used to restore the documents and folders.

tape-device-name: Specify the name of the tape device whose tape files are used to restore the documents and folders. If you are using more than one tape device (up to a maximum of four), specify the names of the devices in the order they are used. If you are using more than one tape volume, you may want to use more than one tape device so one volume is rewound and/or unloaded while another tape device is processing.

SYSOBJNAM Parameter

Specifies the system object names of the documents that are being restored. This parameter is required if OBJ(*SYSOBJNAM) is specified. Only system object names of documents can be specified. A full 10 characters must be specified for the name, and a maximum of 300 names can be specified.

SAVFLR Parameter

Specifies the name of the save folder on the media or in an online save file from which the documents and folders are restored.

***ANY:** The first document that meets the values specified by the DLO parameter is restored, regardless of the folder from which it was saved.

*NONE: The documents being restored were saved as documents without folders. *NONE is valid only when DLO(*ALL) is specified.

save-folder-name: Specify the name of the save folder from which documents and/or folders are being restored. A maximum of 63 characters can be specified for the path name. When DLO(*ALL) is specified, a maximum of 300 folder names can be specified. A save folder name must be specified if DLO(document-name) is specified.

RENAME Parameter

Specifies the new user-assigned name for the restored document.

***SAME:** The documents being restored have the same user-assigned names that they have on the media.

document-name: RENAME specifies the new document names that the documents have after they are is restored. A maximum of 300 user-assigned names can be specified for documents being restored.

TOFLR Parameter

Specifies the name of the folder currently on the system in which the restored documents are placed. A maximum of 63 characters can be specified. If DLO(*SYSOBJNAM) is specified, and the system name of the document without a folder is given, the RSTFLR parameter is not allowed. Also, if SAVFLR(*ANY) or SAVFLR(*NONE) is specified, the RSTFLR parameter is not allowed.

***SAME:** The documents being restored are placed into the same folder that is specified on the SAVFLR parameter.

restore-folder-name: Specify the name of the folder where the restored documents are placed.

NEWOBJ Parameter

Specifies if a new library-assigned name is created for the folder and/or document being restored.

***SAME:** A new library assigned name is not created for the documents or folders being restored.

*NEW: A new library-assigned name is created for each document or folder being restored.

VOL Parameter

Specifies the volume identifier of the tape volumes (reels) or diskette volumes that the documents and folders are restored from. The volumes must be in the same order as they were when the documents and folders were saved. (For more information on the VOL parameter, see Appendix A, "Expanded Parameter Descriptions.")

***MOUNTED:** The documents and folders are to be restored from the volumes that are currently on the device specified by the DEV parameter.

volume-identifier: Specify the identifiers of one or more volumes in the order they are on the device and used to restore the documents and folders. For tape volumes and single-diskette volumes, a maximum of 6 characters identify each volume.

SEQNBR Parameter

Specifies the starting tape used and which sequence number is used for the restore process. This parameter is valid for tape only.

***SEARCH:** The volume on the device is searched for a data file containing the documents to be restored; when a match is found, the data file is searched for the documents to be restored. If the specified documents are not found in that data file, the search continues for the next data file that matches the QDOC label. If the last operation on the device specifies ENDOPT(*LEAVE) (the tape is positioned at the location at which the last operation ended), the file search starts with the first data file beyond the current tape position. If ENDOPT(*LEAVE) is not used for the last operation (or if the tape was manually rewound since an ENDOPT(*LEAVE operation), the search begins with the first data file on the volume.

starting-sequence-number: Specify the starting sequence number of the file used for the restore process.

***ONLY:** The ending sequence number is the same as the starting sequence number.

ending-sequence-number: Specify one of the following to use as the ending sequence number:

LABEL Parameter

Specifies the file label that is used to find the file that was written onto the media during the save operation.

***GEN:** The file label is created by the system. When doing a restore from diskette or tape of folders that have more than 8000 object in them, if the user specified label DLO001 and the user specified label was reset, then the restore will update the label automatically from the next media file with labels DLO002, DLO003, and so forth.

data-file-identifier: Specify the media data file identifier (up to 17 alphameric characters) that contains the documents and folders to restore.

SAVF Parameter

Specifies the name of the save file (library-name/save-file-name) that contains the saved documents and/or folders restored.

The possible library values are:

***LIBL:** The library list is used to locate the save file.

library-name: Specify the library where the save file is located.

ENDOPT Parameter

Specifies, only when tape is used, what operation is automatically performed on the tape volume after the restore operation ends. If more than one reel is included, this parameter applies only to the last reel used; all other reels are rewound and unloaded when the end of the tape is reached.

***REWIND:** The tape is automatically rewound, but not unloaded, after the restore operation has ended.

***LEAVE:** The tape is not rewound; another restore operation can start at the current position on the tape.

***UNLOAD:** The tape is automatically rewound and unloaded after the restore operation has ended.

SAVDATE Parameter

Specifies the date that the documents and folders were saved. If more than one version of the documents exist on the media, enter the date that specifies which version of the documents and folders are restored. (The date must be specified in the system date format specified by the system value QDATFMT; if separators specified by the system value QDATSEP are used, the value must be enclosed in apostrophes.) If SAVDATE is not specified, the version of the documents and folders being restored is the first version found on the volume or the version found with the specified file label.

SAVTIME Parameter

Specifies the time that the documents and folders were saved. Specify the time as a 6-digit value, in the format hours, minutes, seconds (hhmmss), to specify which version of the documents and folders is restored if more than one version of the document exists on the media. If colons are used as separators, the value must be enclosed in apostrophes ('hh:mm:ss'). If a volume identifier is specified, but SAVTIME is not specified, the version of the documents and folders being restored is the first version found on the volume or the version found with the specified file label.

This parameter is valid only if SAVDATE is also specified.

OUTPUT Parameter

Specifies a list is created that shows information about the status of the documents and folders. The listing shows the parameter values that were specified, and the object name and library-assigned document name (LADN) for all documents and folders restored. Information related to the primary owner of each document is also listed for the restored documents and folders.

***NONE:** No list is created.

***PRINT:** The output is printed with the job's spooled output on a printer.

Examples

```
RSTDLO  DLO(*SYSOBJNAM)  DEV(diskette device name)  +
   SYSOBJNAM(HZ83B55219)
```

This command restores the document named HZ83B55219 that was saved.

```
RSTDLO  DLO(A)  DEV(diskette device name)  +
   SAVEFLR(X)
```

This command restores the document named A from folder X.

```
RSTDLO  DLO(*ALL)  DEV(QDKT)
```

This command restores all documents and folders that are currently on QDKT. If an object is on QDKT more than once, then each version of it is restored. The last version restored is the final version.

```
RSTDLO  DLO(*SYSOBJNAM)  DEV(QDKT) +
      SYSOBJNAM(HZ83B55219)  NEWOBJ (*NEW)
```

This command restores document HZ83B55219 and gives it a new library-assigned name and a new system object name.

```
RSTDLO  DLO(A B)  DEV(QDKT) SAVFLR(C) +
       RENAME(Y  Z)  RSTFLR(X)
```

This command restores documents A and B from within folder C. Document A is renamed to Y and document B is renamed to Z and puts them in folder X.

```
RSTDLO  DLO(*ALL)  DEV(tape device name) SAVFLR(A) +
      SEQNBR(1 3)  LABEL(DLO001)
```

If folder A contains 20,000 documents and the SAVDLO command reset the user specified label, then this command restores all of folder A from tape files with the sequence numbers 1, 2, and 3, and the labels DLO001, DLO002, and DLO003.

RSTLIB (Restore Library) Command

The Restore Library (RSTLIB) command restores to the system one or all user libraries that were saved on diskette or tape, or restores a user library that was saved in a save file. Any library that was saved by the Save Library (SAVLIB) command can be restored by this command. The RSTLIB command restores the whole library, which includes the library description, object descriptions, and contents of the objects in the library. (It also includes status information for program temporary fixes (PTFs) that were in the library at the time the library was saved. Note that the restore of a library destroys the PTF status information that currently exists in the library being restored.)

For job queues, message queues, output queues, data queues, logical files, and save files, only the object descriptions are restored, because only the definitions are saved. Also, logical file access paths may be restored if they were saved. More information on restoring access paths is in the *Data Base Guide*.

This command can be used to restore libraries regardless if there are objects in the library that had their storage freed by a command other than the restore's corresponding SAVLIB command, or if any library was deleted by the Delete Library (DLTLIB) command. If the data portions of the objects in the saved libraries were not freed, each library is copied into the same area of storage that it previously occupied. If the storage was freed, the system finds the needed storage to store the library contents (the object description and data portion of every file, program, and journal receiver in the library). If the library does not exist on the system because it has been deleted or the library is being restored on a different system, the system must find the storage to store everything that is in the library, including the library description.

The user profile of the system default owner (QDFTOWN) becomes the default owner of any object restored in the system whose owning profile does not exist on the system.

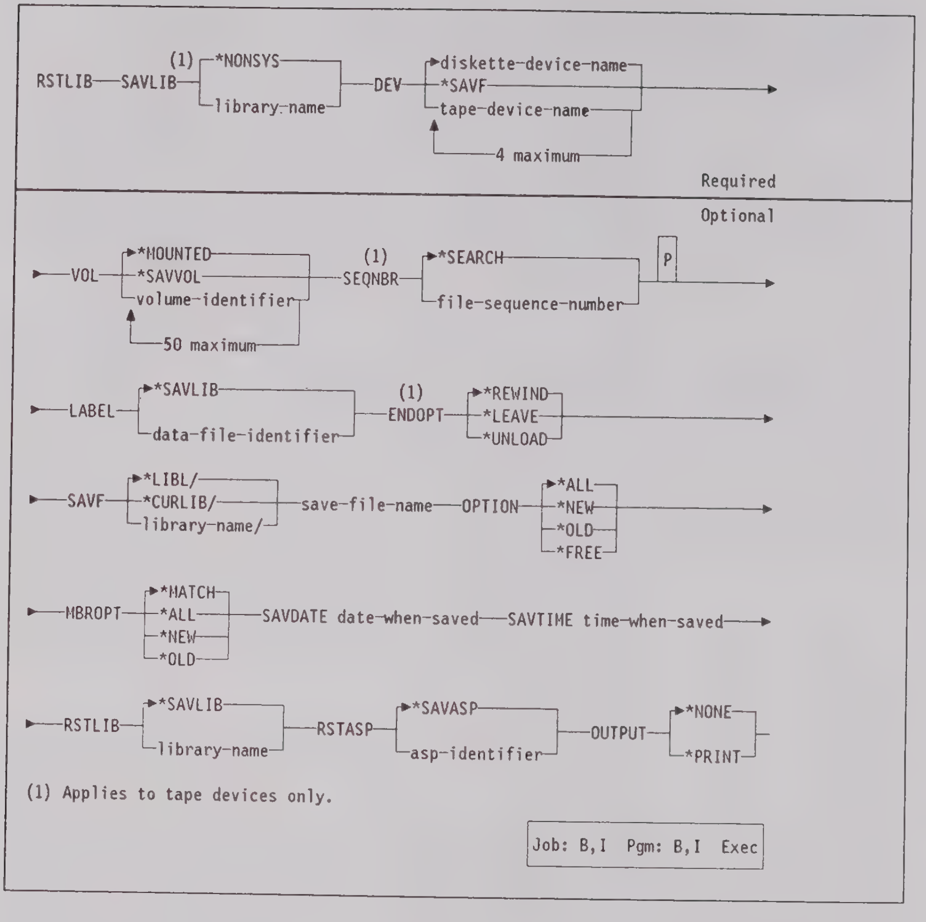
If an object already exists in the library it is being restored in, the public and private authorities of the existing object are kept. If the object does not exist in the library, all public authorities are restored, but any private authorities must be given again. For an existing output queue object that is actively spooling during the restore, or a data queue that already exists in the library, the object is not restored, and a diagnostic message is sent.

Note: You must have authority to the Create Save File (CRTSAVF) command to restore a save file to a library where it does not already exist. The RSTLIB command ignores all file overrides that are currently in effect for the job, with the exception of the overrides for the restore file.

Restrictions: To use this command, you must have either *SAVSYS authority specified in your user profile by the SPCAUT parameter, or you must have all of the following:

1. Read and add authority for, or be the owner of, each library specified.
2. Object existence authority for, or be the owner of, each object in the library if the object already exists in the library on the system (object existence and use authority are required for message queue objects). If the object does not exist, you must have add authority for the user profiles that own each object being restored. (If you do not have the correct authority for all the libraries and objects specified, only those for which you do have the authority are restored.)
3. If VOL(*SAVVOL) is specified, use authority to the saved-from library. Use authority for the save file are required when restoring from a save file. Use authority for the device description and the device file are required when restoring from tape or diskette.

The current versions of programs on the system being restored should not be running while the library is being restored. If any program is running, it ends abnormally.



SAVLIB Parameter

Specifies the name of the library, or all libraries, that are restored in the system. (You cannot specify *NONSYS if the RSTLIB parameter specifies anything other than *SAVLIB.)

***NONSYS:** All libraries saved by the Save Library (SAVLIB) command with LIB(*NONSYS) specified are restored. All other operations on the system must be stopped before this option is specified. This requires ending all subsystems through the End Subsystem (ENDSBS) or End System (ENDSYS) command.

library-name: Specify the name of the library to restore. The name of the library being restored must be the same as that used when the library was saved. The names QSYS, QSRV, QTEMP, QSPL, QDOC, and QRECOVERY cannot be specified.

DEV Parameter

Specifies the name of the device used to restore the library. The device name must already be known on the system by a device description.

diskette-device-name: Specify the name of the diskette device that is used to save or restore the objects.

***SAVF:** The save or restore operation is done using the save file specified by the save file (SAVF) parameter.

tape-device-name: Specify the names of one or more tape devices used for the save or restore. If you are using more than one tape device (up to a maximum of four), specify the names of the devices in the order to be used. If you are using more than one tape volume, you may want to use more than one tape device so that one tape volume can rewind and/or unload while another tape device is processing.

VOL Parameter

Specifies the volume identifiers of the tape volumes (reels) or diskette volumes from which the library is restored. The volumes must be put in the same order as they were when the library was saved. If the library is known to the system before the restore operation, the system can verify if the volumes put on tape contain the current version of the library. A maximum of 50 entries can be specified. (For more information on VOL parameter, refer to Appendix A, "Expanded Parameter Descriptions.")

***MOUNTED:** The library is restored from the volumes that are currently put on the device specified by the DEV parameter. The version of the restored library is the first version found on the media, unless a specific version is identified by the SAVDATE and SAVTIME parameters, or for tape, the SEQNBR parameter.

***SAVVOL:** The system, using the save/restore history information, determines which tape or diskette volumes contain the most recently saved version of the library. If the device type specified in the DEV parameter does not match the device type of the most recently saved version of the library, an error message is sent to the user, and the function is ended. If *SAVVOL is specified, the parameters SAVDATE and SAVTIME cannot be specified. IF *SAVVOL is specified, the SEQNBR parameter can only be *SEARCH.

volume-identifier: Specify the identifiers of one or more volumes in the order they are put on tape and used to restore the libraries.

LABEL Parameter

Specifies the name that identifies the data file on the tape or diskette used for the restore. This label must have been specified on the save command.

***SAVLIB:** The file label is created by the system using the name of the library specified on the SAVLIB parameter.

data-file-identifier: Specify the data file identifier (a maximum of 17 characters) of the data file used for the restore. This option is valid only for single library restores.

SEQNBR Parameter

Specifies, only when tape is used, which sequence number is used for the restore process.

***SEARCH:** The volume put on tape is searched for a data file with an identifier that matches the SAVLIB parameter value; when a match is found, the data file is restored. If the last operation on the device specifies ENDOPT(*LEAVE) (the tape is positioned at the location where the last operation ended), the file search starts with the first data file beyond the current tape position. If ENDOPT(*LEAVE) was not used for the last operation (or if the tape was manually rewound after an ENDOPT(*LEAVE) operation), the search starts with the first data file on the volume.

file-sequence-number: Specify the sequence number of the file used for the restore process.

ENDOPT Parameter

Specifies, only when tape is used, what operation is automatically done on the tape volume after the restore operation ends. If more than one reel is included, this parameter applies only to the last reel used; all other reels are rewound and unloaded when the end of the tape is reached.

***REWIND:** The tape is automatically rewound, but not unloaded, after the restore operation has ended.

***LEAVE:** The tape is not rewound; another restore operation can start at the current position on the tape.

***UNLOAD:** The tape is automatically rewound and unloaded after the restore operation has ended.

SAVF Parameter

Specifies the qualified name of the save file that contains the save data for the library that is restored.

The possible library values are:

***LIBL:** The library list is used to locate the save file.

***CURLIB:** The current library for the job is used to locate the save file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the save file is located.

OPTION Parameter

Specifies if the objects to restore in the saved library are known to the system and how to handle each object in the library.

***ALL:** All the objects in the saved library are restored to the library. Old objects on diskette, tape, or in a save file replace the current versions in the library in the system, and the objects not having a current version are added to the library in the system. Objects presently in the library remain there.

***NEW:** Only the objects in the saved library that do not exist in the current version of the library in the system are added to the library. (Only objects not known to the library in the system are restored; known objects are not restored.) This option restores objects that were deleted after they were saved or that are new to this library. If any saved objects have a version already in the library in the system, they are not restored and an informational message is sent for each one, but the restore continues.

***OLD:** Only the objects in the library having a saved version are restored; that is, the version of each object currently in the library is replaced by the saved version. (Only objects known to the library are restored.) If any saved objects are no longer part of the on-line version of the library, they are not added to the library and an informational message is sent for each one, but the restore continues.

***FREE:** The saved objects are restored only if they exist in the library in the system with their space freed. The saved version of each object is restored in the system in its previously freed space. (This option restores objects that had their space freed when they were saved.) If any saved objects are no longer part of the current version of the library, or if the space is not free for any object, the object is not restored and an informational message is sent for each one. The restore continues, and all of the freed objects are correctly restored.

MBROPT Parameter

Specifies, for data base files already known to the system, which members are restored. Unless MBROPT(*MATCH) is used, the member list in the saved file need not match, member for member, the current version in the system.

Note: Before restoring a file, the system verifies that the file and member creation dates of the known system objects are a match for the creation dates of the objects to be restored. If verification fails, the file is not restored.

<u>*MATCH:</u> The saved members are restored if the lists of the files where they exist are a match, member for member, for the lists of the current system version.

***ALL:** All members in the saved file are restored.

***NEW:** Only new members (members not known to the system) are restored.

***OLD:** Only members already known to the system are restored.

SAVDATE Parameter

Specifies the date the library was saved. If the most recently saved version is not restored, or if more than one saved version is on the volumes, enter the date that specifies which version of the library is restored. (The date must be entered in the system date format specified by the system value QDATFMT; if separators, specified by the system value QDATSEP, are used, the value must be enclosed in apostrophes.) If a volume identifier or VOL(*MOUNTED) is specified, but SAVDATE is not, the version of the library to be restored is the first version found on the volume. This parameter is valid only if a volume identifier or VOL(*MOUNTED) is specified, or if a save file (SAVF parameter) is specified.

SAVTIME Parameter

Specifies the time that the library was saved if the current version is not restored. Specify the time as a six-digit value, in the format hours, minutes, and seconds (hhmmss), that specifies which version of the library to restore. If colons are used as separators, the value must be enclosed in apostrophes ('hh:mm:ss'). If a volume identifier or VOL(*MOUNTED) is specified, but SAVTIME is not, the version of the library to be restored is the first version found on the volume.

This parameter is valid only if SAVDATE is also specified.

RSTLIB Parameter

Specifies if the library contents are restored to the same library they were saved in, or to a different one. (If a different library is specified, you cannot specify *NONSYS on the SAVLIB parameter.)

<u>*SAVLIB:</u> The library contents are restored to the same library or libraries they were saved in.

library-name: Specify the name of the library where the saved library contents are restored. The names QSYS, QSRV, QRECOVERY, and QTEMP *can* be specified. If *NONSYS is specified on the SAVLIB parameter, a library name cannot be specified in this parameter.

RSTASP Parameter

Specifies if the objects are restored to the same auxiliary storage pool (ASP) from which they were saved or to another specific one. The only objects that can be restored to a user ASP (2 through 16) are journals, journal receivers, and save files. All other objects are restored to the auxiliary storage pool (ASP). This parameter applies to all libraries specified.

<u>*SAVASP:</u> The objects are restored to the same auxiliary storage pools from which they were saved.

asp-identifier: When the specified ASP is 1, the objects are restored to the system ASP, and when the ASP is 2 through 16, the objects are restored to the user ASP specified.

OUTPUT Parameter

Specifies if a list that shows information about the status of the objects is created. The list shows the parameter values that were specified and shows all objects, restored and not restored. Information relative to each object's security is listed for the restored objects.

<u>*NONE:</u> No output is created.

***PRINT:** The output is supplied with the job's spooled output on a printer.

Examples

```
RSTLIB  SAVLIB(JOE)  DEV(TAP1)  OPTION(*NEW)
```

This command restores the saved version of library JOE from tape device TAP1. The only objects that are restored in the library are new objects (ones that were in the library when they were saved and then deleted later).

```
RSTLIB  SAVLIB(*NONSYS)  DEV(TAP1)  OUTPUT(*PRINT)
```

This command restores all the saved user-created libraries, QGPL, and licensed program libraries, such as QRPG and QIDU, to the system from tape. The contents of the libraries are restored as they were saved. New objects (on tape) are added to the system; old objects in the system are overlaid by the version of the old objects on tape. Because OUTPUT(*PRINT) is specified, a printout of all objects, restored and not restored for each library, is sent with the job's spooled output to the printer. Each library after the first starts a new page. After each library, a completion message states how many objects were restored and how many were not restored. At the end of a list, a final completion message states how many libraries were restored and not restored.

```
RSTLIB  SAVLIB(PAYROLL)  DEV(TAP1)  +
  SAVDATE(060188)  SAVTIME(103214)  RSTLIB(OLDPAY)
```

This command restores the version of the PAYROLL library from the device TAP1, whose volume identifier is PAY. The version to be restored was saved at 10:32:14 on the date 06/01/88. All of the objects in the saved PAYROLL library are restored to the library OLDPAY.

```
RSTLIB  SAVLIB(QGPL)  DEV(PAY1)  VOL(QGPL  QGPL)
```

This command restores the QGPL library from two tape volumes both named QGPL. Even though the volume identifiers are the same, they must both be specified. (If the tape volumes are put in the wrong order, an error message is sent to the system operator message queue.)

```
RSTLIB  SAVLIB(USRLIB)  DEV(TAP1 TAP2 TAP3)  +
  VOL(USRA USRB USRC USRD)  ENDOPT(*UNLOAD)
```

This command restores library USRLIB from four volumes on three tape devices. Volume USRA is put on tape device TAP1, volume USRB on TAP2, volume USRC on TAP3, and volume USRD on TAP1. The operator removes volume USRA from TAP1, so that TAP1 can be used by volume USRD.

```
RSTLIB  SAVLIB(LIB1)  DEV(TAP1)  +
  MBROPT(*ALL)  SAVDATE(082388)  SAVTIME(123251)  +
  RSTLIB(LIB2)  OUTPUT(*PRINT)
```

This command restores the version of library LIB1 from the device TAP1. The version to be restored was saved at 12:32:51 on the date 08/23/88. All of the objects in the saved library LIB1 are restored to library LIB2. A list of restored objects and those not restored is given.

```
RSTLIB  SAVLIB(LIB1)  DEV(*SAVF)  SAVF(SAVF1)
```

This command restores library LIB1 from the save file SAVF1.

```
RSTLIB  SAVLIB(LIB1)  DEV(*SAVF)  SAVF(SAVF1)  RSTASP(2)
```

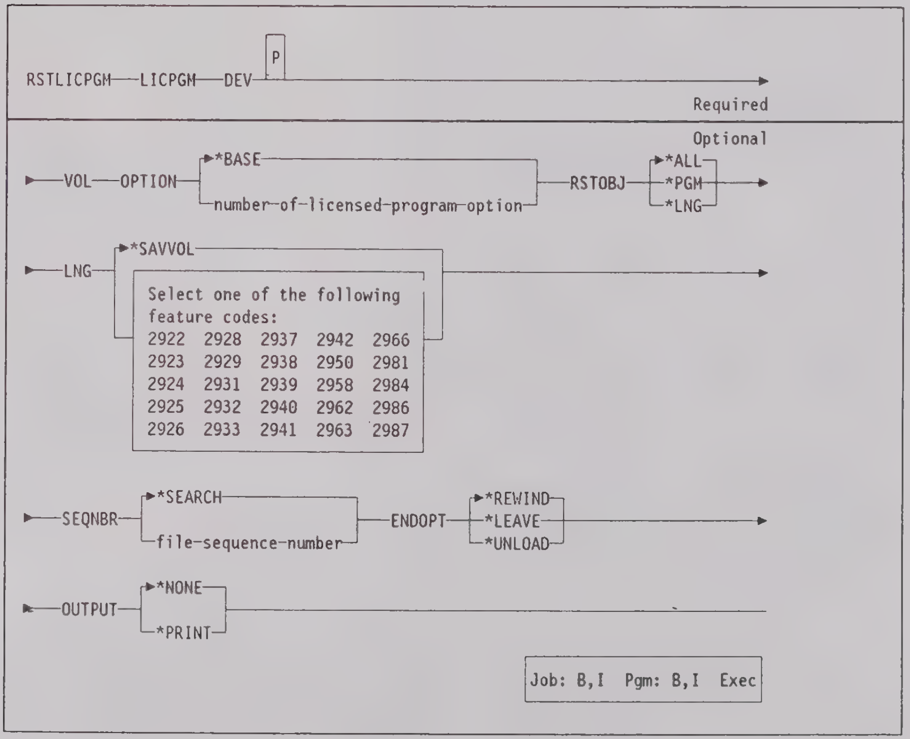
This command restores library LIB1 from the save file SAVF1. Any journals, journal receivers, and save files in the saved version of library LIB1 are restored to auxiliary storage pool 2 (unless they already exist in library LIB1 and are in a different ASP).

RSTLICPGM (Restore Licensed Program) Command

The Restore Licensed Program (RSTLICPGM) command loads or restores an IBM-supplied licensed program from a specified library for initial installing or new-release installing.

The processing of the command is determined by the contents of a data area that is restored at the start of the command. In addition, with respect to the data area, additional programs can be called during the processing of the RSTLICPGM command. These programs are included with the licensed program and are necessary to complete the restore of the licensed program.

Restrictions: (1) You must have authority to the RSTLICPGM command. (2) If this command is used to restore a program in the licensed program, the copy of the program that is currently in the system should not be running while the program is being restored. If this occurs, the processing program is ended abnormally. (3) If other objects of the licensed program are in use, they are not restored. (4) With the exception of overrides for the restore printing OUTPUT(*PRINT), this command ignores all file overrides that are currently in effect for the job. (5) Some licensed programs can only be restored if you are enrolled in the system distribution directory. See the publications for each licensed program for a description of this restriction.



LICPGM Parameter

Specifies the seven-character identifier of the licensed program that is restored.

5728AP1: The advanced double-byte character printer set licensed program is deleted.

5728BA1: The BASIC licensed program is deleted.

5728CB1: The COBOL licensed program is deleted.

5728CM1: The communication licensed program is deleted.

5728CR1: The cryptographic support licensed program is deleted.

5728DB1: The S/38 utilities licensed program is deleted.

5728DCT: The language dictionary licensed program is deleted.

5728DS1: The graphic support licensed program is deleted.

5728PC1: The personal computer (PC) licensed program is deleted.

5728PL1: The PL/I licensed program is deleted.

5728PS1: The PASCAL licensed program is deleted.

5728PT1: The advanced performance tools licensed program is deleted.

5728PW1: The programming development aids licensed program is deleted.

5728QU1: The Query licensed program is deleted.

5728RG1: The RPG licensed program is deleted.

5728SS1: The optional part of the operating system specified in the OPTION parameter is deleted.

5728ST1: The Structured query language program is deleted.

5728WP1: The office support licensed program is deleted.

DEV Parameter

Specifies the name of the device used for the restore of the licensed program. The device name must already be known on the system by a device description. If multiple devices are specified, they must have compatible media formats.

VOL Parameter

Specifies the volume identifiers of the tape volumes or diskette volumes from which the licensed program is restored. The volumes must be put on the device in the same order as they were when the licensed program was saved. A maximum of 50 entries can be specified. (For more information on the VOL parameter, refer to Appendix A, "Expanded Parameter Descriptions.")

OPTION Parameter

Specifies which licensed program option to restore.

***BASE:** Specifies only the base part of the licensed program is restored.

number-of-licensed-program-option: Specify the number associated with the optional part of the listed licensed program that is to be restored.

RSTOBJ Parameter

Specifies what type of licensed program objects is restored.

***ALL:** All of the objects for the licensed program are restored. This includes both the program objects and the language objects as specified by the LNG parameter.

The RSTOBJ(*ALL) value should be used when the save of the licensed program has been done with the SAVLICPGM command such that the language objects follow the program objects immediately on the media. If used on the distribution tape where the language objects and program objects have been saved separately, the restore searches the distribution media currently on the device for the appropriate file. This may cause the tape to be read to the end-of-volume.

***PGM:** Only the program objects for the licensed program are to be restored. The RSTOBJ(*PGM) value should be used when restoring the program objects from the distribution media where the program objects and the selected language objects are not on the same tape.

***LNG:** The objects associated with the primary language identified by the language feature indicated on the LNG parameter are restored.

The RSTOBJ(*LNG) value should be used when changing the primary language for a product.

LNG Parameter

Indicates the language for the licensed program.

***SAVVOL:** If the language feature for the licensed program on the save media matches the system language feature, the language objects are restored to the licensed program's libraries. If the language features do not match, the language objects are restored into the multilingual library for that language feature.

feature codes: Indicates the feature ID of the primary language for the licensed program. More information on feature identifications can be found in the *Licensed Programs Installation Guide.*

SEQNBR Parameter

Specifies which sequence number to use for the restore process.

***SEARCH:** The volume on the device is searched for a data file for the licensed program; when a match is found, the objects are restored. If the last operation on the device specifies ENDOPT(*LEAVE), indicating that the tape is positioned at the location where the last operation ended, the file search starts with the first data file beyond the current tape position. If ENDOPT(*LEAVE) was not used for the last operation, or if the tape was manually rewound since an ENDOPT(*LEAVE) operation, the search starts with the first data file on the volume.

file-sequence-number: Specify the sequence number of the file used for the restore process.

ENDOPT Parameter

Specifies what operation is automatically done on the tape volume after the restore operation ends. If more than one reel is included, this parameter applies only to the last reel used; all other reels are rewound and unloaded when the end of tape is reached.

***REWIND:** The tape is automatically rewound, but not unloaded, after the restore operation has ended.

***LEAVE:** The tape is not rewound. Another restore operation can start at the current position on the tape.

***UNLOAD:** The tape is automatically rewound and unloaded after the restore operation has ended.

OUTPUT Parameter
Specifies if output showing information about the status of the licensed program is created. The output shows all objects, restored and not restored. Information relative to each object's security is listed for the restored objects.

***NONE:** No output is created.

***PRINT:** The output is provided with the job's spooled output on a printer.

Example

```
RSTLICPGM  LICPGM(5714BA1)  DEV(TAP1)
```

This command restores the BASIC licensed program to the system. The licensed program objects must be put on the TAP1 tape drive. Because no other parameters are specified, the defaults are used for the command.

RSTOBJ (Restore Object) Command

The Restore Object (RSTOBJ) command restores to the system a single object, or a group of objects, in a single library that were saved on diskette, tape, or in a save file by a single command. The types of objects that can be restored by this command are listed in the OBJTYPE parameter. They can be saved either as separate objects or as part of the library save. The RSTOBJ command restores the object description and contents of each object specified in the command.

For job queues, output queues, user-defined message queues, logical files, save files, and data queues, only the object descriptions are restored; the contents of those objects are not restored. However, logical file access paths can be saved by a save command with ACCPTH(*YES) specified; if so, the access paths can be restored. More information on restoring access paths is in the *Data Base Guide*.

The command can be used to restore the objects even if the object storage was freed by the Save Object (SAVOBJ) or Save Library (SAVLIB) command, or if the objects were deleted by the associated delete command for that object type. If the storage was not freed, each object is restored in the same area of storage that it previously occupied. If the version of the object being restored is larger than the version in the system (for example, data records that were deleted from the system still exist in the saved version of a file), the additional storage needed for the object is acquired. If the saved version of the object is smaller (for example, data records that are added to the system), the space that was acquired for the object remains assigned to that object and is available for use by the object.

If logical file access paths were saved (ACCPTH(*YES) was specified when the objects were saved), the access paths are restored if (1) all based-on physical files are also being restored by the same restore command, (2) the logical file is also being restored by the same restore command, or the logical file already exists on the system (the same file exists, not a recreated version), and (3) MAINT(*IMMED or *DLY) is in effect for the logical file (if it still exists on the system).

If the storage was freed, the system finds the needed storage to store the contents of each file, journal receiver, and program (only the data portion). If the objects do not exist on the system because they were deleted or they are being restored in a different system, the system must find the storage to store everything about each unknown object (its description and data portion).

The user profile of the system default owner (QDFTOWN) becomes the default owner of any objects restored in the system whose owner is not known to the system.

If an object already exists in the library to which it is restored, the public and private authorities of the existing object are kept. If the object does not exist in the library, all public authorities are restored, but any private authorities must be given again. For an existing output queue object that is actively spooling during the restore process, or a data queue that already exists in the library, none of the attributes are restored and a diagnostic message is issued.

Restrictions: (1) You must have use authority for the Create Save File (CRTSAVF) command when restoring a save file that does not currently exist on the system. (2) With the exception of overrides for the restore listing file, this command ignores all file overrides that are currently in effect for the job. (3) To use this command, you must have either the special authority *SAVSYS specified in your user profile (SPCAUT parameter), or you must have:

1. Read and add authority for, or be the owner of, the specified library.
2. Object existence authority for, or be the owner of, each object specified if the object already exists in the library on the system (object existence and use authority are required for message queue objects). If the object does not already exist in the library on the system, you must have add authority for the user profiles that own each object being restored.

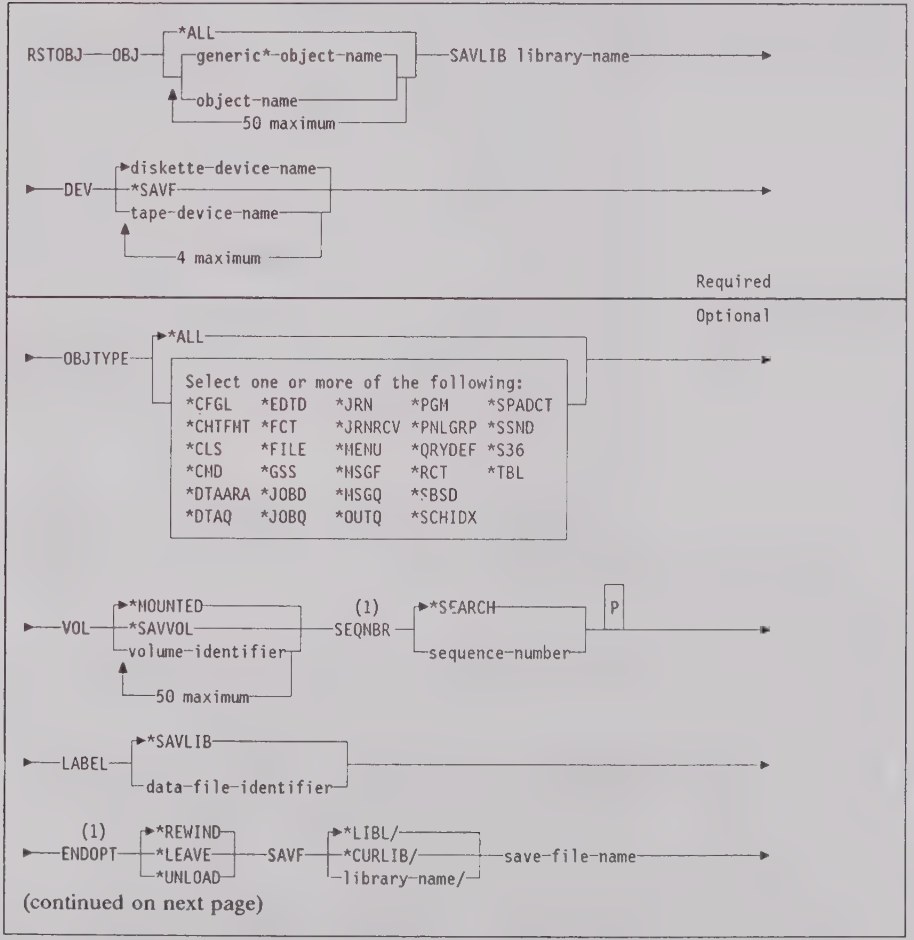
3. If VOL(*SAVVOL) is specified, use authority to the saved-from library.

(4) The RSTOBJ command does not restore the library's data dictionary or its associated data base files. To do this, the RSTLIB command must be used.

Use authority for the save file is required when restoring from a save file. Use authority is required to the device description when restoring from tape or diskette.

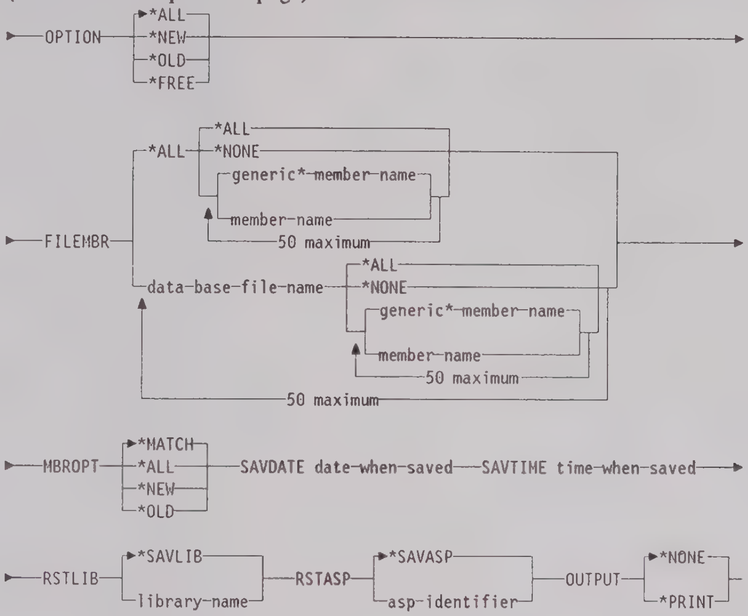
If this command is used to restore a program, the copy of that program that is currently in the system should not be running while the program is being restored. If this occurs, the running program is abnormally ended.

Objects saved by separate commands must also be restored by separate commands. If a single command is used, some of the objects are not restored.



(continued on next page)

(continued from previous page)



(1) Applies to tape devices only.

Job: B,I Pgm: B,I Exec

OBJ Parameter

Specifies the names of one or more objects to restore. All the objects must be in the library specified in the SAVLIB parameter. Specify the name of each object or the generic name of each group of objects to restore. (Add an asterisk after the last character in the generic name; for example, ABC*. If an * is not included with the name, the system assumes that the name is a complete object name.)

If the OBJTYPE parameter is not specified when the command is specified, all the object types listed in the description of the OBJTYPE parameter are restored, if they are in the specified library on the media or in the save file and have the specified names.

***ALL:** All the objects saved from the specified library are restored, depending on the values specified for OBJTYPE and OPTION.

generic-object-name:* Specify one or more generic names of groups of objects in the specified library to restore.

object-name: Specify one or more names of specific objects to restore. (Both generic names and specific names can be specified in the same command.)

SAVLIB Parameter

Specifies the name of the library that contained the objects when they were saved. If RSTLIB is not specified, this is also the name of the library to which the objects are restored. Specify the name of the library.

DEV Parameter

Specifies the name of the device used to restore the objects. The device name must already be known on the system by a device description.

diskette-device-name: Specify the name of the diskette devices used to save or restore objects.

***SAVF:** The save or restore operation is done using the save specified on the save file (SAVF) parameter.

tape-device-name: Specify the name of one or more tape devices that restore the objects. If you are using more than one tape device (up to a maximum of four), specify the names of the devices in the order to be used. If using more than one tape volume, you may want to use more than one tape device, so that one tape volume can rewind and/or unload while another tape device is processing.

OBJTYPE Parameter

Specifies the types of OS/400 objects that are restored.

***ALL:** All object types that are specified by name and were saved from the specified library are restored. If *ALL is also specified for the OBJ parameter, then all objects saved for that library are restored.

object-type: Specify the value for each of the types of objects that are restored. The following types can be specified.

Value	Type of OS/400 Object
*CFGL	Configuration list
*CHTFMT	Chart format
*CLS	Class
*CMD	Command
*DTAARA	Data area
*DTAQ	Data queue
*EDTD	Edit description
*FCT	Forms control table
*FILE	File
*GSS	Graphics symbol set
*JOBD	Job description
*JOBQ	Job queue
*JRN	Journal
*JRNRCV	Journal receiver
*MENU	Menu definition
*MSGF	Message file
*MSGQ	Message queue (user-defined only)
*OUTQ	Output queue
*PGM	Program
*PNLGRP	Panel group
*QRYDEF	Query definition
*RCT	Reference code translation table
*SBSD	Subsystem description
*SCHIDX	Search information search index
*SPADCT	Spelling aid dictionary
*SSND	Session description
*S36	System/36 machine description
*TBL	Table

The object types shown are also the ones that are saved and restored by the Save Library (SAVLIB), Save Change Object (SAVCHGOBJ), Save Object (SAVOBJ), Restore Library (RSTLIB), and Restore Licensed Program (RSTLICPGM) commands. Data dictionaries (*DTADCT) are only restored with RSTLIB. Other object types can be saved only by the Save System (SAVSYS) command and restored by the OS/400 install operation, or saved by the Save Document Library Object (SAVDLO) command and restored by the Restore Document Library Object (RSTDLO).

VOL Parameter

Specifies the volume identifiers of the tape volumes or diskette volumes from which the objects are restored. The volumes must be put on tape in the same order as they were when the objects were saved. If the objects are known to the system before the restore operation, the system can verify if the volumes put on tape contain the current version of each object. A maximum of 50 entries can be specified. (For an more information on the VOL parameter, refer to Appendix A, "Expanded Parameter Descriptions.")

***MOUNTED:** The objects are restored from the volumes that are currently put on the device specified by the DEV parameter. The version of the objects that is restored is the first version found in the specified location, unless a specific version is identified by the SAVDATE and SAVTIME parameters.

***SAVVOL:** The system, using the save/restore history information, determines which tape or diskette volumes contain the most recently saved version of the objects. When *SAVVOL is specified, the following operational characteristics and restrictions apply:

- If the characteristics of the device specified in the DEV parameter do not match the device and location of the most recently saved version of the library, an error message is sent to the *user*, and the function is ended.
- If the wrong volume is put on a device in the location specified by the command, a message is sent to the *system operator* that identifies the first volume that must be put on a device before the objects can be restored.
- When OBJ(*ALL) and OBJTYPE(*ALL) is specified, the objects are considered in the order they would appear in a display produced by the Display Library (DSPLIB) command. (That is, the object names and types specified in the RSTOBJ command are used to determine which file of saved objects is used in the restore operation. One file is produced for each SAVLIB or SAVOBJ command run.) The file chosen is the one in which the first considered object was last saved. Each object considered that either was not saved in the file chosen to be processed, or that was more recently saved, is not restored; for each object not restored, an error message is sent to the user.
- If *SAVVOL is specified, the parameters SAVDATE and SAVTIME cannot be specified.
- If *SAVVOL is specified and the RSTLIB value is equal to the SAVLIB value, OPTION(*NEW) cannot be specified.
- If *SAVVOL is specified, the SEQNBR parameter can only be *SEARCH, and LABEL can only be *SAVLIB.

volume-identifier: Specify the identifiers of one or more volumes in the order in which they are put on a device and used to restore the objects.

SEQNBR Parameter

Specifies, only when tape is used, which sequence number is used for the restore process.

***SEARCH:** The volume put on a device is searched for a data file with an identifier that is a match for the SAVLIB parameter value; when a match is found, the object is restored. If the last operation on the device specifies

ENDOPT(*LEAVE) (the tape is positioned at the location where the last operation ended), the file search starts with the first data file beyond the current tape position. If ENDOPT(*LEAVE) was not used for the last operation (or if the tape was manually rewound after an ENDOPT(*LEAVE) operation), the search starts with the first data file on the volume.

sequence-number: Specify the sequence number of the file used for the restore process.

LABEL Parameter

Specifies the name that identifies the data file on the tape or diskette used for the restore. This label must have been specified on the save command.

***SAVLIB:** The file label is created by the system using the name of the library specified on the SAVLIB parameter.

data-file-identifier: Specify the data file identifier (a maximum of 17 characters) of the data file used for the restore.

ENDOPT Parameter

Specifies, only when tape is used, what operation is automatically done on the tape volume after the save or restore operation ends. If more than one reel is included, this parameter applies only to the last reel used; all other reels are rewound and unloaded when the end of tape is reached.

***REWIND:** The tape is automatically rewound, but not unloaded, after the save or restore operation has ended.

***LEAVE:** The tape is not rewound; another save or restore operation can start at the current position on the tape.

***UNLOAD:** The tape is automatically rewound and unloaded after the save or restore operation has ended.

SAVF Parameter

Specifies the qualified name of the save file that contains the save data for the objects that are restored.

The possible library values are:

***LIBL:** The library list is used to locate the file.

***CURLIB:** The current library for the job is used to locate the file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the file is located.

OPTION Parameter

Specifies if each saved object to restore is known to the library in the system that the objects are being restored to and how to handle the objects.

***ALL:** All the objects in the saved library are restored in the system. Old objects on diskette, tape, or in a save file, replace the current versions in the library in the system, and the objects not having a current version are added to the library in the system. Objects presently in the library remain there.

***NEW:** Only the objects in the saved library that do not exist in the current version of the library in the system are added to the library. (Only objects not known to the library in the system are restored; known objects are not restored.) This option adds objects that were deleted after they were saved or that are new to this library. If any saved objects have a current version in the library in the system, they are not restored and an informational message is sent for each one, but the restore continues.

***OLD:** Only the objects in the library that have a saved version are restored; that is, the current version of each object is replaced by the saved version. (Only objects known to the library are restored.) If any saved objects are no longer part of the current version of the library, they are not added to the library and an informational message is sent for each one, but the restore continues.

***FREE:** The saved objects are restored only if they exist in the library in the system with their space freed. The saved version of each object is restored in the system in its previously freed space. (This option restores objects that had their space freed when they were saved.) If any saved objects are no longer part of the current version of the library, or if the space is not free for any object, the object is not restored and an informational message is sent for each one. The restore continues and all of the freed objects are correctly restored.

FILEMBR Parameter

Specifies the data base file members to restore. This parameter is made up of two parts: the file name and the member name. This parameter cannot be specified if MBROPT(*MATCH) is specified.

The file name values are:

***ALL:** The member list following this value applies to all files specified on the OBJ parameter.

data-base-file-name: Specify the name of the data base file that contains the members to be restored. There may be up to 50 of the file/member combinations specified for a single command.

Restrictions:

- Each data base file specified on the FILEMBR parameter must also be specified on the OBJ parameter by either its complete name, a generic name, or *ALL.
- The OBJTYPE parameter must be *ALL, or it must include *FILE.
- Generic names are not valid for the data base file name, but are allowed for the member name.

- Duplicate file names are not allowed.

The member name values are:

***ALL:** All members are saved or restored from the specified file.

***NONE:** No members are saved or restored from the specified file.

generic-member-name:* Specify the generic name of the member to be restored from the specified file.

member-name: Specify the specific name of the member to be restored from the given file.

Restrictions:

- If the nongeneric member names are specified, all specified members must exist in the file in order for any part of the file to be saved or restored.
- If generic member names are specified, the file is not saved if it does not contain any members that match the generic names. (For example, if you specify PAY* as a generic member name, and the system is unable to find any members whose name starts with PAY, the file is not saved.) If no files specified by the FILEMBR parameter are saved because no members with the specific generic name can be found, a diagnostic message is sent, the restore ends, and an escape message is sent specifying the number of files not saved. If at least one of the files processed for the FILEMBR parameter is saved, the diagnostic message is not sent, and the number of files not saved is displayed on the final completion message.
- The members are first checked using the MBROPT parameter. If a match is found, then they are checked against the list of the FILEMBR parameter.

MBROPT Parameter

Specifies, for data base files that already exist on the system, which members to restore. Unless MBROPT(*MATCH) is used, the member list in the saved file need not match, member for member, the current version in the system.

Note: Before restoring a file, the system checks to ensure that the file and member creation dates of the known system objects match the creation dates of the objects to be restored. If this check fails, the file is not restored.

***MATCH:** The saved members are restored if the lists of the files where they reside match, member for member, the lists of the current system version.

***ALL:** All members in the saved file are restored.

***NEW:** Only new members (members not existing on the system) are restored.

***OLD:** Only members already existing on the system are restored.

SAVDATE Parameter

Specifies the date the objects were saved. If the most recently saved version is not restored, or if multiple saved versions reside on the media volume, specify the date that specifies which version of the objects to restore. (The date must be specified in the system date format specified by the system value QDATFMT; if separators, specified by the system value QDATSEP, are used, the value must be enclosed in apostrophes.) If a volume identifier or VOL(*MOUNTED) is specified, but SAVDATE is not, the version of the objects to be restored is the first version found on the volume. This parameter is valid only if a volume identifier or VOL(*MOUNTED) is specified, or if a save file (SAVF parameter) is specified.

SAVTIME Parameter

Specifies the time that the objects were saved. Specify the time as a six-digit value, in the format hours, minutes, and seconds (hhmmss), that specifies which version of the objects to restore if the current version is not restored. If colons are used as separators, the value must be enclosed in apostrophes ('hh:mm:ss'). If a volume identifier or VOL(*MOUNTED) is specified, but SAVTIME is not, the version of the objects to be restored is the first version found on the volume.

This parameter is valid only if SAVDATE is also specified.

RSTLIB Parameter

Specifies if the objects are restored to a different library or to the same library where they were saved.

***SAVLIB:** The objects are restored to the same library where they were saved.

library-name: Specify the name of the library to which the saved objects are restored.

RSTASP Parameter

Specifies if the objects are restored to the same auxiliary storage pool where they were saved or to another specific one. The only objects that can be restored to a user ASP(2 through 16) are journals, journal receivers, and save files. All other objects are restored to the system ASP.

***SAVASP:** The objects are restored to the same auxiliary storage pools where they were saved.

asp-identifier: When the specified ASP is 1, the objects are restored to the system ASP, and when the ASP is 2 through 16, the objects are restored to the user ASP specified.

OUTPUT Parameter

Specifies if a list that shows information about the status of the objects is created. The list shows the parameter values that were specified and shows all objects, restored and not restored. Information relative to each object's security is listed for the restored objects.

<u>*NONE:</u> No output is created.

***PRINT:** The output is provided with the job's spooled output on a printer.

Examples

```
RSTOBJ   OBJ(PAYROLL)  SAVLIB(LIBX)  DEV(TAP1) OBJTYPE(*PGM) VOL(*SAVVOL)
```

This command restores to LIBX the program named PAYROLL that was saved from LIBX. The TAP1 tape drive is used to restore the most recently saved version of the program.

```
RSTOBJ   OBJ(PAY*)  SAVLIB(LIBX)  DEV(DKT1)  VOL(ABCD)  +
  OPTION(*OLD)  SAVDATE(102288)  SAVTIME(143000)  +
  RSTLIB(LIBY)
```

All objects whose names start with PAY and that were saved from library LIBX on diskette volume ABCD at 14:30:00 on 10/22/88 are restored to LIBY. Volume ABCD must be put on diskette device DKT1. Because OPTION(*OLD) is specified, the only objects restored are those having the same object name and type both in LIBY in the system and LIBX on diskette.

```
RSTOBJ   OBJ(NEWPROG)  SAVLIB(QGPL)  DEV(DKT1)  OBJTYPE(*PGM)  +
  VOL(PGMS)  OPTION(*NEW)
```

A new program named NEWPROG is added to the general purpose library, QGPL. It is restored from a volume labeled PGMS that is inserted in the diskette device DKT1.

```
RSTOBJ   SAVLIB(LIB)  DEV(TAP1)  OBJTYPE(*PGM)  VOL(SVOL)  +
  SEQNBR(2)  SAVDATE(082388)  SAVTIME(143000)  OUTPUT(*PRINT)
```

All programs that were saved from library LIB that exist in the saved version on tape volume SVOL, sequence number 2, saved at 14:30:00 on 08/23/88 are restored to library LIB. An output list of all objects restored as well as objects not restored is provided.

```
RSTOBJ OBJ(*ALL) SAVLIB(BACKUP) DEV(*SAVF)  +
  OBJTYPE(*JRNRCV) SAVF(SAVEJ) RSTASP(3)
```

All journal receivers that were saved from library BACKUP into save file SAVEJ are restored to the library BACKUP. The journal receivers are restored to auxiliary storage pool 3 (unless they already exist in library BACKUP and are in a different ASP).

RSTS36F (Restore System/36 File) Command

The Restore System/36 File (RSTS36F) command reads a file, creates a data base physical or logical file (if it doesn't already exist), and copies any data into the file. The input file can be a diskette file, tape file, or a data base physical file on the AS/400[2] system. The file could have been created on a System/36 using the SAVE system OCL procedure (or the equivalent OCL call of the $COPY SSP utility), or on an AS/400 using the SAVS36F CL command.

The RSTS36F command accepts a diskette file created on a System/36 as a compressed file.

The RSTS36F command accepts a diskette or tape file created on a System/34 or System/32 using the $COPY utility. System/34 or System/32 diskette offline multiple-volume files are not accepted by this command.

If the file being restored does not exist in the library specified on the TOLIB parameter, it is created. A physical file member is added using the name syntax 'Myymmdd', which identifies the original creation date of the file. This naming convention is needed by the System/36 environment in order to support date-differentiated files.

If the restore function creates the file and the file was not previously secured, the new file is owned by the user issuing the RSTS36F command and the file is created with a default authority of *ALL (that is the same as "AUT(*ALL)").

It should be noted that this function is intended only for exchanging files with a System/36.

Restrictions:

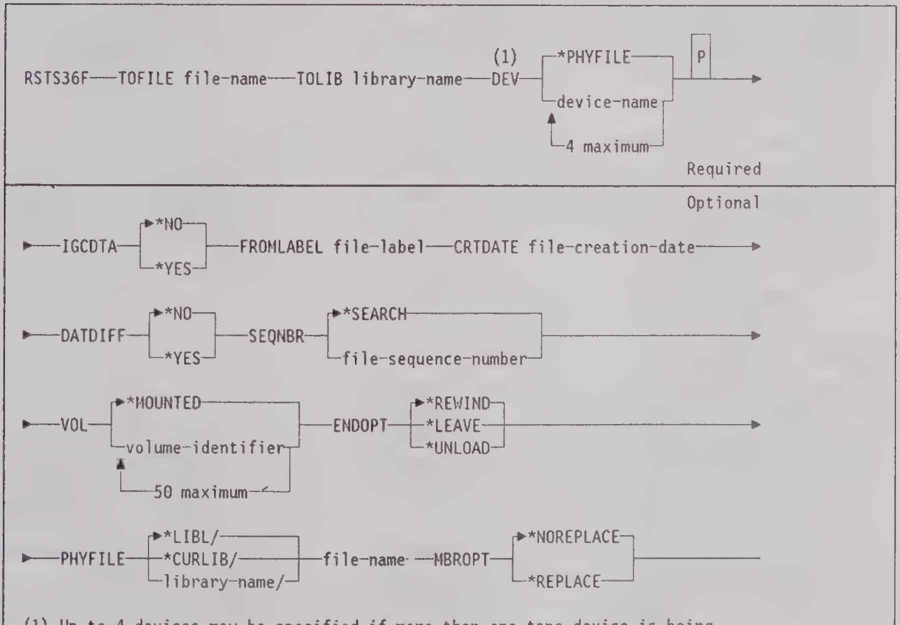
- The following authorities are required (normally only applies when running on a system using resource security):
  1. *USE authority for this command.
  2. *USE authority for the library specified in the TOLIB parameter.
  3. *CHANGE authority for the library specified in the TOLIB parameter when restoring a file that does not already exist on the system.
  4. *CHANGE and *OBJMGMT authority for the existing file (needed to add a new member) when restoring a date-differentiated physical file and a file by the same name but with a different creation date.
  5. *ALL authority for the file when restoring to an existing physical file with the same creation date and MBROPT(*REPLACE) specified.
  6. *CHANGE authority for the based-on physical file (this physical file was referred to as the *parent file* on System/36) if the file being restored is a System/36 alternative index file (that is, a logical file).
  7. *USE authority for the diskette device description object, *USE authority for device file QSYSDKT in library QSYS, when restoring from diskette
  8. *USE authority for the tape device description object, *USE authority for device file QSYSTAP in library QSYS, when restoring from tape.
  9. *USE authority for the file, and *USE authority for the library that contains the file (PHYFILE parameter) if restoring from a data base physical file
  10. If the file doesn't exist on the system but a file authorization holder object by the same name does exist, *ALL authority or ownership for the authorization holder object.
- There is no replace function supported when restoring System/36 alternative index (that is logical) files. If restoring an alternative index file, no file object by the same name can already exist in the specified library.
- If restoring a logical file, the based-on physical file must already exist in the specified library (TOLIB parameter).

---

[2] AS/400 is a trademark of the International Business Machines Corporation.

- The name specified for the TOFILE parameter must meet AS/400 naming standards.
- The AS/400 files that are the same as the System/36 date-differentiated files are multiple-member physical files. Date-differentiated alternative index files are not supported. The data for a physical file is stored in a member named using the syntax 'Myymmdd' where 'yymmdd' represents the original creation date (in year/month/day format) of the file.

  Because all members of a physical file share the same file attributes (for example, record length and key information), date-differentiated files by the same name restored to the same library are required to have the same file attributes. If an attribute mismatch is present, the files are not restored.
- Object-level and record-level functions, including read operations, should not be used for a file being restored by RSTS36F. Activity running at the same time against the file (for example, moving the file or reading or adding records) can cause the restore function to stop if the file cannot be allocated.



(1) Up to 4 devices may be specified if more than one tape device is being used. (Only one device is allowed with diskette.)

Job: B,I Pgm: B,I Exec

TOFILE Parameter
Specifies the name given to the file when it is restored to the system. This parameter allows the file to be renamed at the time it is restored to the system.

TOLIB Parameter
Specifies which library should contain the file being restored from the input file.

DEV Parameter
Specifies the names of the devices used for the restore operation. Each device name must already be known on the system by a device description.

***PHYFILE:** The input file is the data base physical file specified by the PHYFILE parameter.

device-name: Specify the name of the diskette device or the names of one or more tape devices that are used for the restore operation. If more than one tape device is used, enter the names of the devices in the order in which they are used. When more than one tape volume is used, it may be advantageous to use more than one tape device so that rewinding and unloading of tape volumes can occur at the same time with processing on another tape device. A maximum of four device names can be specified.

IGCDTA Parameter
Specifies if the file being restored can contain double-byte character set (DBCS) data.

*NO: The file being restored may not contain DBCS data. If a file already exists by the name specified on the TOFILE parameter in the specified library and the file allows DBCS data, an informational messages is sent and the restore continues. The resulting file will allow DBCS data (the file's IGC attributes is not changed by the restore).

*YES: The file being restored may contain DBCS data. If a file already exists by the name specified on the TOFILE parameter in the specified library and the file does not allow DBCS data, an informational messages is sent and the restore continues. The resulting file will not allow DBCS data (the file's IGC attributes is not changed by the restore).

FROMLABEL Parameter

Specifies the label of the diskette or tape file that contains the file being copied to the AS/400 (it identifies the input file for the restore process). A maximum of 8 characters can be used to label the diskette or tape file.

If no value is specified, the value of the TOFILE parameter will be used as the diskette or tape label. If the TOFILE parameter is an extended name, the characters between the quotes will be used as the label. For example, specifying TOFILE("MIKE&BOB") and no FROMLABEL parameter would be the same as specifying FROMLABEL('MIKE&BOB').

CRTDATE Parameter

Specifies the creation date of the diskette or tape file used for the restore operation. (The date must be entered in the job date format (DATFMT); if separators, specified by the job date separator character (DATSEP), are used, the value must be enclosed in apostrophes.)

For diskettes coming from a System/36, there may be multiple files on one diskette with the same name and different creation dates. Specifying a value for the CRTDATE parameter is the only way of choosing which diskette file to use.

For tapes from System/36 or AS/400, there can be multiple files on a tape with the same name. The user can use the CRTDATE or SEQNBR parameter to choose the correct file. If a numeric SEQNBR parameter and the CRTDATE parameter are specified, the SEQNBR parameter takes precedence (if the file at the specified sequence number has a different creation date, the restore function ends).

Specify the creation date of the tape or diskette file used for the restore. The date specified is changed to Julian format (cyyddd) for tape and international format (yymmdd) for diskette.

If no value is specified, the diskette or tape file creation date is not used to select the input file. For diskette, take the first file in the diskette VTOC with the name specified on the FROMLABEL parameter. For tape, take the first file on the tape with the name specified on the FROMLABEL parameter. Note that if the previous tape operation had specified ENDOPT(*LEAVE), the search for a matching file does not start at the start of the tape reel.

DATDIFF Parameter

Specifies if the restore should allow date-differentiated files (that is, multiple files with the same name but different file creation dates).

*NO: Date-differentiated files are not allowed:

- If a file already exists by the name specified on the TOFILE parameter in the specified library, and the file contains a member with a different creation date (as determined by the member name), an error message is sent and the file is not restored.
- If a file already exists by the name specified on the TOFILE parameter in the specified library, and the file contains a member with the same creation date (as determined by the member name), either the data in that member is replaced (if MBROPT(*REPLACE) is specified) or an error message is sent (if MBROPT(*NOREPLACE) is specified).
- If no file exists by the name specified on the TOFILE parameter in the specified library, the file is restored normally.

*YES: Date-differentiated files are allowed:

- If a file already exists by the name specified on the TOFILE parameter in the specified library, and the file contains a member with a different creation date (as determined by the member name), a member is added to the file (only if the file attributes match), and the file's data is restored into the new member.
- If a file already exists by the name specified on the TOFILE parameter in the specified library, and the file contains a member with the same creation date (as determined by the member name), either the data in that member is replaced (if MBROPT(*REPLACE) is specified) or an error message is sent (if MBROPT(*NOREPLACE) is specified).
- If no file exists by the name specified on the TOFILE parameter in the specified library, the file is restored normally.

SEQNBR Parameter

Specifies, only when tape is used, which sequence number is used for the restore process.

*SEARCH: The volume on the device is searched for a data file with an identifier that matches the FROMLABEL parameter value; when a match is found, the data file is restored. If the last operation on the tape device specified ENDOPT(*LEAVE), the file search starts with the data file at the current tape position. Otherwise, the search starts with the first data file on the tape volume.

file-sequence-number: Specify the sequence number of the tape file used for the restore process. The header label name from the file found by the SEQNBR parameter is compared to the name specified for the FROMLABEL parameter. If the names are the same, restore processing can continue. If the names are different, the restore function ends.

VOL Parameter

Specifies the volume identifiers of the tapes or diskettes used for restoring the file. If a diskette or tape volume label does not match the specified volume identifier, an inquiry message is sent which allows the user to Cancel (stop the restore), Ignore (go ahead and use the mounted diskette or tape), or Retry (make the volume check again on a newly mounted diskette or tape).

***MOUNTED:** The file being restored is copied from whatever volume is on the device. For diskettes, use whatever volume is loaded in the diskette device.

volume-identifier: Specify the volume identifiers of the tapes or diskettes used for restoring the file. For each tape or diskette volume name, a maximum of 6 characters can be specified using any combination of letters and digits. A maximum of 50 volume names can be specified.

ENDOPT Parameter

Specifies, only when tape is used, what operation is automatically performed on the tape volume after the restore operation ends. If more than one reel is involved, this parameter applies only to the last reel used. (This parameter is ignored if a diskette device is specified in the DEV parameter.)

***REWIND:** The tape is automatically rewound, but not unloaded, after the restore operation has ended.

***LEAVE:** The tape is not rewound; another tape operation can start at the current position on the tape.

***UNLOAD:** The tape is automatically rewound and unloaded after the restore operation has ended.

PHYFILE Parameter

Specifies the qualified name of the data base physical file that is used as the input file for the restore process.

The file must have a record length of 256.

If the specified file does not exist or is not a physical file, the file is not restored. If the physical file contains multiple members, the first member of the file is used.

The possible library values are:

***LIBL:** The library list is used to locate the physical file.

***CURLIB:** The current library for the job is used to locate the physical file. If no current library entry exists in the library list, QGPL is used to locate the file.

library-name: Specify the library where the physical file is located.

file-name: Specify the name of the data base file that is used as the input file for the restore process.

MBROPT Parameter

Specifies if the data of an existing physical file member is replaced.

***NOREPLACE:** If a file already exists by the name specified on the TOFILE parameter in the specified library, and the file contains a member with the same creation date (as determined by the member name), an error message is sent and the data in that member is *not* replaced.

***REPLACE:** If a file already exists by the name specified on the TOFILE parameter in the specified library, and the file contains a member with the same creation date as the file being restored, the data in that member of the file *is* replaced.

Examples

```
RSTS36F  TOFILE(ACCTRCV)  TOLIB(QS36F)  DEV(I1)  +
  CRTDATE('01/22/85')  VOL(SAVE1)
```

This command restores the file ACCTRCV into library QS36F. Assuming that I1 is the name of a diskette device description object, the file is restored from the diskette placed in the diskette device. The diskette must have a volume name of SAVE1. The diskette file used for the restore must have a file label of ACCTRCV and a creation date of January 22, 1985 (assuming the job date format is *MDY and the date separator is a '/').

```
RSTS36F  TOFILE(PAY.VIEW)  TOLIB(PAYLIB)  DEV(T1)  +
  FROMLABEL('P*V')  ENDOPT(*LEAVE)
```

The file P*V is restored from device T1 as a file named PAY.VIEW in library PAYLIB. Assuming T1 is a tape device, the file is copied from one or more tapes that are on device T1. No check is made on the tape volume name. When the restore ends, the tape is left positioned at the end of tape file P*V.

```
RSTS36F  TOFILE(ACCTPAY)  TOLIB(QS36F)  DEV(*PHYFILE)  +
  PHYFILE(NETLIB/SENDFILE)
```

This command restores the file ACCTPAY in library QS36F from physical file SENDFILE in library NETLIB.

RSTS36FLR (Restore System/36 Folder) Command

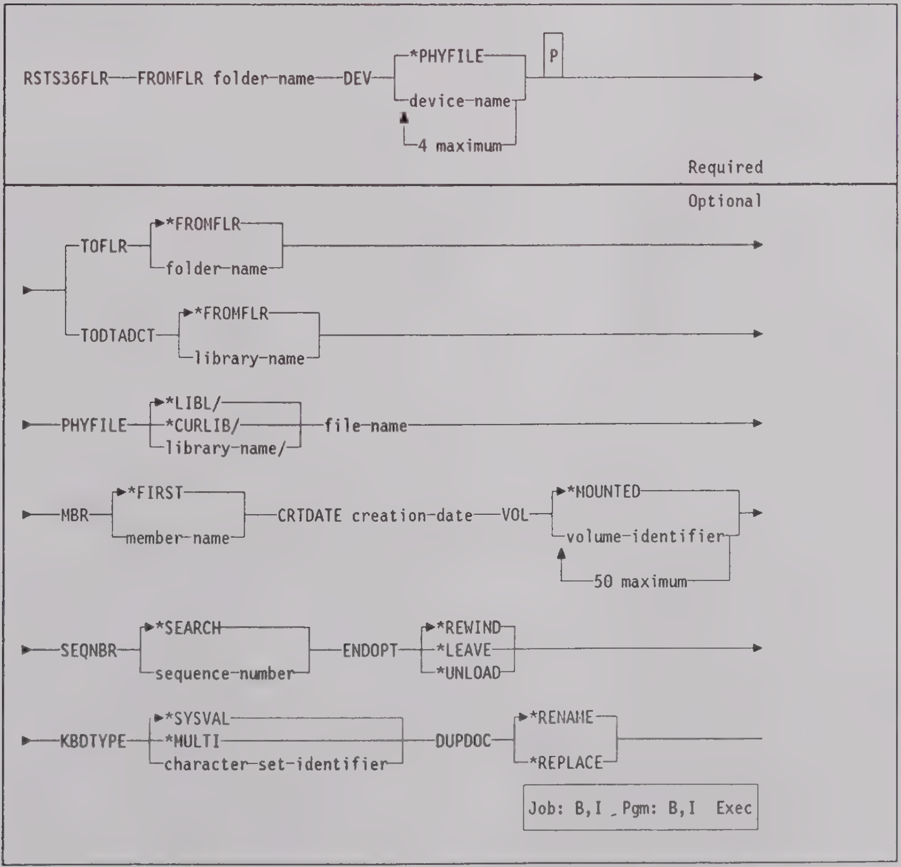
The Restore System/36 Folder (RSTS36FLR) command reads a diskette, tape, or data base file which was prepared by the System/36 SAVEFLDR command and converts the data so that it can be used by the AS/400[3] system.

Note: When restoring a folder from the System/36 that contains an invalid name for the AS/400 system, the name is automatically changed to one generated by the system.

Two types of System/36 folders are supported as input; document folders and interactive data definition utility (IDDU) folders.

- Document folders are converted into AS/400 folders and the documents contained in them are likewise converted into AS/400 documents. If a document on System/36 was in the process of being edited when the SAVEFLDR was done, there would have been two existing versions of the document: an unchanged original, and a second one containing modifications. When this folder is restored on the AS/400 system, both the unchanged original and the modification version will be created. If you complete the editing using the office function on the AS/400 system, the modification version will be removed.
- System/36 IDDU folders do not create folders on the AS/400 system. Instead, a library is created and the data dictionary is converted into an AS/400 data dictionary.

Restriction: This command supports only document folders and IDDU folders.



---

[3] AS/400 is a trademark of the International Business Machines Corporation.

FROMFLR Parameter

Specifies the name of the folder that you want restored.

DEV Parameter

Specifies the name of the device that you want to restore the specified folder from. The device name must be known to the system by a device description.

***PHYFILE:** The folder is in a data base file. If DEV(*PHYFILE) is specified, then the PHYFILE parameter must be specified.

device-name: Specify the name of the tape or diskette device to use to restore the folder. A maximum of 4 devices for tape can be specified.

TOFLR Parameter

Specifies the name of the folder that you want the restored folder's contents put in. This parameter is only used for folders that are document folders. An error message is sent if this parameter is specified for any other type of folder.

***FROMFLR:** The old name of the folder that is specified in the FROMFLR parameter is used for the new name of the folder when it is restored.

folder-name: Specify the new name for the folder being restored.

TODTADCT Parameter

Specifies the name of the library where an IDDU dictionary (folder) will be restored. This parameter is only used for IDDU dictionaries. An error message is sent if this parameter is specified for any other type of folder. When this parameter is specified, a library is created and a data dictionary is created in the new library. An error message is sent if a library by the specified name already exists.

***FROMFLR:** The old name of the folder that is specified in the FROMFLR parameter is used for the new library name when it is restored.

library-name: Specify the name of the library where you want to put the folder being restored.

PHYFILE Parameter

Specifies the qualified name of the data base file that contains the System/36 folder.

The possible library values are:

***LIBL:** The library list is used to locate the file.

***CURLIB:** The current library for the job is used to locate the file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library that the file is located in.

| *file-name:* Specify the name of the data base file that contains the System/36 folder.

MBR Parameter

Specifies the name of the member to restore.

***FIRST:** The first member in the file is restored.

member-name: Specify the name of the member that contains the System/36 folder.

CRTDATE Parameter

Specifies the creation date of the diskette or tape file used for the restore operation. (The date must be entered in the job date format (DATFMT); if separators, specified by the job date separator character (DATSEP), are used, the value must be enclosed in apostrophes.)

For diskettes coming from a System/36, there may be multiple files on one diskette with the same name and different creation dates. Specifying a value for the CRTDATE parameter is the only way of choosing which diskette file to use.

For tapes from System/36 or AS/400, there can be multiple files on a tape with the same name. The user can use the CRTDATE or SEQNBR parameter to choose the correct file. If a numeric SEQNBR parameter and the CRTDATE parameter are specified, the SEQNBR parameter takes precedence (if the file at the specified sequence number has a different creation date, the restore function ends).

Specify the creation date of the tape or diskette file used for the restore. The date specified is changed to Julian format (cyyddd) for tape and international format (yymmdd) for diskette.

VOL Parameter

Specifies the volume identifiers to the tape volumes (reels) or diskette volumes that the objects will be restored from. The volumes must be put on the device in the same order as they were when the folders were saved.

***MOUNTED:** The tape or diskette that is on the specified device is used.

volume-identifier: Specify the identifier of one or more volumes in the order that you will use them. A maximum of six characters identify each volume.

SEQNBR Parameter

Specifies the sequence number of the file that you want restored.

***SEARCH:** The tape is searched for the first sequence number that contains the specified folder.

sequence-number: Specify the sequence number of the file that you want restored.

ENDOPT Parameter

Specifies, only when tape is used, what operation is automatically performed on the tape volume after the restore operation ends. If more than one reel is involved, this parameter applies only to the last reel used. (This parameter is ignored if a diskette device is specified in the DEV parameter.)

***REWIND:** The tape is automatically rewound, but not unloaded, after the restore operation has ended.

***LEAVE:** The tape is not rewound; another restore operation can begin at the current position on the tape.

***UNLOAD:** The tape is automatically rewound and unloaded after the restore operation has ended.

KBDTYPE Parameter

Specifies the character set and code page of the System/36 that created the data in the folder.

***SYSVAL:** The System/36 from which the folder was saved was the same character set and code page as the AS/400 system.

***MULTI:** The System/36 was configured as a multinational system.

character-set-identifier: If the System/36 was a non-multinational system and the AS/400 system is not the same character set and code page as the System/36, then specify one of the following identifiers. On the System/36, a list of language or country names was presented and you selected from that. If you chose multinational, the System/36 mapped it into a single code regardless of language. For example, if you chose Denmark and not multinational, you should use DMB with RSTS36FLR. If you chose Denmark with multinational on the System/36, you should use *MULTI with RSTS36FLR.

Character Translation Table

Language/Country	Identifier
Arabic X/Basic	CLB
Austria/Germany	AGB
Belgium	BLB
Canadian French	CAB
Denmark	DMB
France (AZERTY)	FAB
France (QWERTY)	FQB
Greece	GKB
Hebrew	NCB
Iceland	ICB
International	INB
Italy	ITB
Japan English	JEB
Japan Katakana	KAB (For 5251, 5291, 5292, and 3180 Katakana display stations)
Netherlands	NEB
Norway	NWB
Portugal	PRB
Latin 2	ROB
Spain	SPB
Spanish Speaking	SSB
Sweden/Finland	SWB
Switzerland/French	SFB
Switzerland/German	SGB
Thailand	THB
Turkey	TKB
United Kingdom	UKB
United States/Canada	USB
Yugoslavia	YGI

DUPDOC Parameter

Specifies if existing documents are renamed or replaced. This parameter is ignored when restoring a data dictionary.

***RENAME:** If a folder already exists by the name specified in the TOFLR parameter and a document in that folder exists by the same name as an incoming document, the incoming document is created in the folder with a new system-generated name. All other documents are created in the specified folder.

***REPLACE:** If a folder already exists by the name specified in the TOFLR parameter, and a document in that folder exists by the same name as an incoming document, the existing document is replaced. All other documents are created in the specified folder.

Examples

```
RSTS36FLR  FROMFLR(TXTDRA)  DEV(DKT01)  +
  CRTDATE('01/16/88')  VOL(SAVE1)
```

This command restores the folder TXTDRA. Assuming that DKT01 is the name of a diskette device description, the folder is restored from the diskette placed in the diskette device. The diskette must have a volume name of SAVE1 and the folder must have a creation date of January 16, 1988.

```
RSTS36FLR  FROMFLR(TXTMEL)  DEV(TAP1)  TOFLR(TXTNEW)  +
  SEQNBR(7)  ENDOPT(*UNLOAD)
```

This command restores the folder TXTMEL to a folder named TXTNEW. Assuming TAP1 is a tape device, the folder is restored from one or more tapes that are on device TAP1. The folder must have a sequence number of 7. The tape will be unloaded from the device at the end of processing.

```
RSTS36FLR  FROMFLR(DONDCT)  DEV(*PHYFILE)  TODTADCT(NEWDCT)  +
  PHYFILE(MYLIB/MYFILE)  MBR(THISONE)
```

This command restores the folder DONDCT to a data dictionary named NEWDCT. Assuming that MYFILE is a physical file located in library MYLIB and contains the member THISONE, the folder is restored from member THISONE.

RSTS36LIBM (Restore System/36 Library Members) Command

The Restore System/36 Library Members (RSTS36LIBM) command reads a file containing library members, creates data base source or data file members on the AS/400 system, and copies the member data from the file into each restored member. The input file can be a diskette file, tape file, or a data base physical file on the AS/400 system. The file could have been created on a System/36, a System/34, or a System/32 using either the FROMLIBR or SAVELIBR system OCL procedure (or the equivalent call of $MAINT), or on an AS/400 system using the SAVS36LIBM CL command.

Diskette files created on a System/34 using the BACKUP procedure or $BACK utility are not accepted by this command. Compressed SAVELIBR diskette files (used by IBM to distribute system libraries for System/36 after release 5.0) are not accepted by this command.

In System/36 terms, the input file format could be a SAVELIBR diskette or tape file, a record-mode LIBRFILE diskette or tape or physical file, or a sector-mode LIBRFILE diskette or tape or physical file. In other words, the input file can be any diskette, tape, or physical file created by the System/36 $MAINT SSP utility.

If the library identified by the TOLIB parameter value does not exist, it is created. Also, the source files QS36SRC and QS36PRC are created if they do not exist in the restore-to library.

If the restore function creates the library, the new library is owned by the user running the RSTS36LIBM command and the library is created with a default authority of *ALL (that is, the same as *AUT(*ALL)*).

If restoring a sector-mode FROMLIBR file or a SAVELIBR file created on a System/36, data files QS36LOD and QS36SBR may also be created to hold restored load and subroutine members. Any restored load and subroutine members are not converted on the current system.

Restrictions:

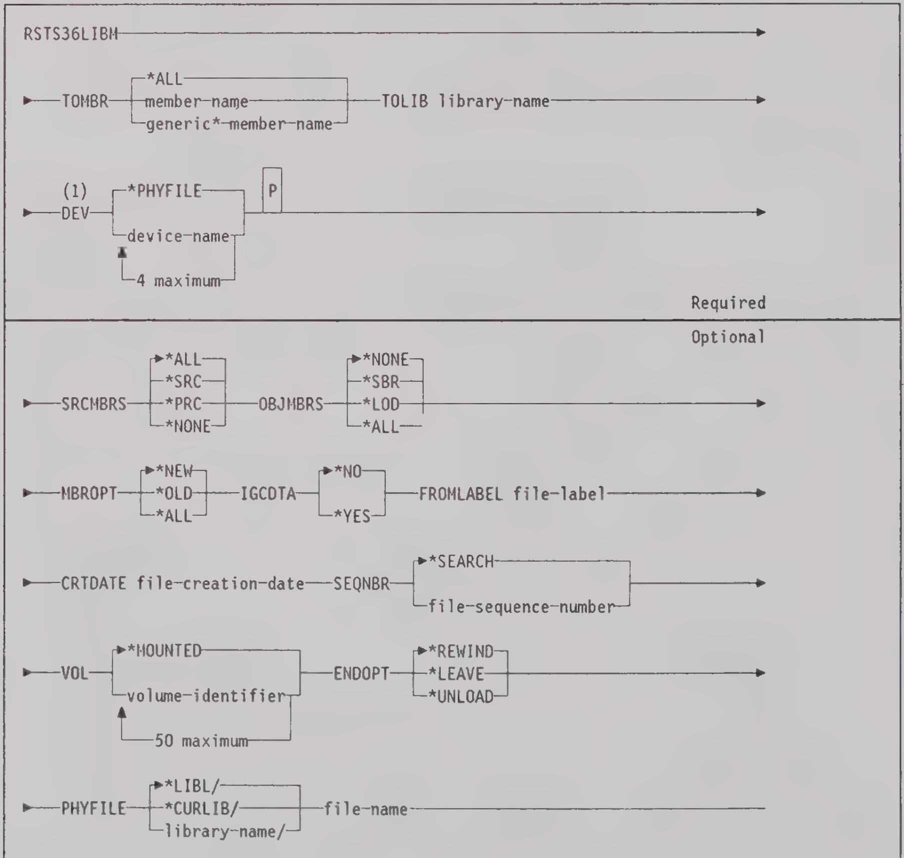
- The following authorities are required (normally only applies when running on a system using resource security):
    1. *USE authority for this command and *USE authority for the Create Library (CRTLIB) command if the library needs to be created.
    2. *USE authority for the CRTSRCPF command if QS36SRC or QS36PRC to be created.
    3. *USE authority for the CRTPF command if QS36LOD or QS36SBR needs creating.
    4. *CHANGE authority for the library specified in the TOLIB parameter.
    5. *CHANGE and *OBJMGMT authority for file QS36SRC in the specified library if restoring source library members.
    6. *CHANGE and *OBJMGMT authority for file QS36PRC in the specified library if restoring procedure library members.
    7. *CHANGE and *OBJMGMT authority for file QS36LOD in the specified library if restoring load library members.
    8. *CHANGE and *OBJMGMT authority for file QS36SBR in the specified library if restoring subroutine library members.
    9. *USE authority for the diskette device description object, *USE authority for device file QSYSDKT in library QSYS if restoring from diskette.
    10. *USE authority for the tape device description object, *USE authority for device file QSYSTAP in library QSYS if restoring from tape.
    11. *USE authority for the file, and *USE authority for the library that contains the file (PHYFILE parameter) if restoring from a data base physical file.
- No object-level or record-level functions should be active for files QS36SRC, QS36PRC, QS36SBR, and QS36LOD while members are being restored by RSTS36LIBM. If the necessary file(s) cannot be allocated exclusively, no members are restored.

- The member name or generic member name specified (TOMBR parameter) must meet AS/400 naming standards.

  If a generic member name or *ALL is specified, a member may be selected for restore that has a name containing characters not allowed in an AS/400 *simple* object name. In this case, the member name will be restored using the AS/400 *extended* name syntax (for example, A!B would become *'A!B'*).

If the name contains a blank, a single quote, a double quote, an asterisk, a question mark, or a device control character (hexadecimal '00'-'3F' or hexadecimal 'FF'), these characters are replaced by underlines and the member is restored using the resulting simple or extended name (for example, A*/? would become *A_/_* and A? would become *A_*).

An informational message is sent each time invalid characters are replaced to get a valid name. An additional informational message is sent if the resulting name change caused a member to be replaced. No message is sent if a member is restored using the extended name syntax without replacing invalid characters.



(1) Up to four devices may be specified if more than one tape device is being used. (Only one device is allowed with diskette.)

Job: B,I Pgm: B,I Exec

TOMBR Parameter

Specifies the names of the members to restore.

*ALL: All members of the specified member types are restored.

member-name: The members having the specified member name are restored.

generic-member-name:* All members which have the specified generic member name are restored.

TOLIB Parameter
Specifies which library should contain the members to restore from the input file.

DEV Parameter
Specifies the names of the devices to use for the restore operation. Each device name must already be known on the system by a device description.

***PHYFILE:** The input file is the data base physical file specified by the PHYFILE parameter.

device-name: Specify the name of the diskette device or the names of one or more tape devices that are used for the restore operation. If more than one tape device is used, specify the names of the devices in the order that they are used. When more than one tape volume is used, it may be advantageous to use more than one tape device so that rewinding and unloading of tape volumes can occur at the same time with processing on another tape device. Specify a maximum of four device names.

SRCMBRS Parameter
Specifies which source member types (source and procedure members on System/36) are restored.

***ALL:** Restore all source and procedure members in the input file that match the member name specified (TOMBR parameter). For example, if you specify TOMBR(PAY*) and SRCMBRS(*ALL), all source and procedure members that start with the letters 'PAY' are restored.

***SRC:** Restore only System/36 source members (to file QS36SRC) that match the member name specified (TOMBR parameter).

***PRC:** Restore only System/36 OCL procedure members (to file QS36PRC) that match the member name specified (TOMBR parameter).

***NONE:** Do not restore any source or procedure library members.

OBJMBRS Parameter
Specifies which object member types (load and subroutine members on System/36) are restored. Because the System/36 and the AS/400 system are not object compatible, any restored members are not immediately useable after the restore. IBM-supplied commands or user-written functions must be run to convert the object member to a useable AS/400 object.

***NONE:** Do not restore any System/36 object members.

***SBR:** Restore only System/36 subroutine members (to file QS36SBR) which match the member name specified (TOMBR parameter).

***LOD:** Restore only System/36 load members (to file QS36LOD) that match the member name specified (TOMBR parameter).

***ALL:** Restore all load and subroutine members in the input file that match the member name specified (TOMBR parameter). For example, if you specify TOMBR(CONF1) and OBJMBRS(*ALL), load and subroutine members with the member name 'CONF1' are restored.

MBROPT Parameter
Specifies which members are restored from the input file if there is not a member by the same name already existing in the corresponding file on the AS/400 system.

***NEW:** Only new members (members which do not already exist in the appropriate source or data file) are restored.

***OLD:** Only old members (members which already exist in the appropriate file) are restored. The existing member is replaced by the copy of the member restored from the file.

***ALL:** All members are restored, even if they already exist in an AS/400 source or data file. Members that do not already exist are created, and members that do exist are replaced.

IGCDTA Parameter
Specifies if source and procedure members being restored can contain double-byte character set (DBCS) data. This attribute is used if the restore operation needs to create source files QS36SRC and QS36PRC to hold the restored library members.

Note: If the QS36SRC or QS36PRC source file already exists in the library specified on the TOLIB parameter and the file's DBCS capability does not match the parameter, an error message is sent and no member is restored.

***NO:** The source or procedure members being restored cannot contain DBCS data.

***YES:** The source or procedure members being restored can contain DBCS data.

FROMLABEL Parameter
Specifies the label (8 characters maximum) of the diskette or tape file that contains the members to copy to the AS/400 system (it identifies the input file for the restore process). If the DEV parameter value is not *PHYFILE, a FROMLABEL value must be specified.

CRTDATE Parameter
Specifies the creation date of the diskette or tape file to use for the restore operation. (The date must be entered in the job date format (DATFMT); if separators are used, specified by the job date separator character (DATSEP), the value must be enclosed in apostrophes.)

For diskettes coming from a System/36, there may be multiple files on one diskette with the same name and different creation dates. Specifying a value for the CRTDATE parameter is the only way of choosing which diskette file to use.

For tapes from System/36 or the AS/400 system, there can be multiple files on a tape with the same name. The user can use the CRTDATE or SEQNBR parameter to choose the correct file. If a numeric SEQNBR parameter and the CRTDATE parameter are specified, the SEQNBR parameter takes precedence (if the file at the specified sequence number has a different creation date, the restore function ends).

Specify the creation date of the tape or diskette file to use for the restore. The date specified is changed to Julian format (cyyddd) for tape and international format (yymmdd) for diskette.

If no value is specified, the diskette or tape file creation date is not used to select the input file. For diskette, take the first file in the diskette VTOC with the name specified on the FROMLABEL parameter. For tape, take the first file on the tape with the name specified on the FROMLABEL parameter. Note that if the previous tape operation had specified ENDOPT(*LEAVE), the search for a matching file will not begin at the start of the tape reel.

SEQNBR Parameter

Specifies, only when tape is used, which sequence number is used for the restore process.

***SEARCH:** The volume that is on the device is searched for a data file with an identifier that matches the FROMLABEL parameter value; when a match is found, the data file is restored. If the last operation on the tape device specified ENDOPT(*LEAVE), the file search begins with the data file at the current tape position. Otherwise, the search begins with the first data file on the tape volume.

file-sequence-number: Specify the sequence number of the tape file to use for the restore process. The header label name from the file found by the SEQNBR parameter is compared to the name specified for the FROMLABEL parameter. If the names are the same, restore processing can continue. If the names are different, the restore function ends.

VOL Parameter

Specifies the volume identifiers of the tapes or diskettes used for restoring the file. If a diskette or tape volume label does not match the specified volume identifier, an inquiry message is sent which allows the user to Cancel (end the restore), Ignore (go ahead and use the diskette or tape that is on the device), or Retry (make the volume check again on the diskette or tape that has just been put on the device).

***MOUNTED:** The file that is restored is copied from whatever volume is on the device. For diskettes, use whatever volume is loaded in the diskette device.

volume-identifier: Specify the volume identifiers of the tapes or diskettes to be used for restoring the file. For each tape or diskette volume name, a maximum of 6 characters can be specified using any combination of letters and digits. A maximum of 50 volume identifiers can be specified.

ENDOPT Parameter

Specifies, only when tape is used, what operation is automatically performed on the tape volume after the restore operation ends. If more than one reel is involved, this parameter applies only to the last reel used. (This parameter is ignored if a diskette device is specified in the DEV parameter.)

***REWIND:** The tape is automatically rewound, but not unloaded, after the restore operation has ended.

***LEAVE:** The tape is not rewound; another tape operation can begin at the current position on the tape.

***UNLOAD:** The tape is automatically rewound and unloaded after the restore operation has ended.

PHYFILE Parameter

Specifies the qualified name of the data base physical file that is used as the input file for the restore process.

The file must have a record length of between 40 and 120 if it contains members saved in record mode. The record length must be 256 if it contains members saved on a System/36 in sector mode.

If the specified file does not exist or is not a physical file, no library members are restored. If the physical file contains multiple members, the first member of the file is used.

The possible library values are:

***LIBL:** The library list is used to locate the physical file.

***CURLIB:** The current library for the job is used to locate the physical file. If no current library entry exists in the library list, QGPL is used to locate the file.

library-name: Specify the library where the physical file is located.

file-name: Specify the name of the data base physical file that is used as the input file for the restore process.

Examples

```
RSTS36LIBM  TOMBR(XYZ1)  TOLIB(JOHNSON)  DEV(I1)  SRCMBRS(*PRC)  +
  MBROPT(*ALL)  FROMLABEL('XYZ1')
```

The single OCL procedure member XYZ1 is restored as a member of source file QS36PRC in library JOHNSON. Assuming I1 refers to a diskette device, the input diskette file must have the label XYZ1.

```
RSTS36LIBM  TOMBR(X*)  TOLIB(ORDER)  DEV(*PHYFILE)  +
  PHYFILE(NETLIB/S36SRC)
```

All source and procedure members with names starting with the character 'X' and that do not already exist as members of QS36SRC and QS36PRC in library ORDER are restored. The members are restored from file S36SRC in library NETLIB.

RSTUSRPRF (Restore User Profiles) Command

The Restore User Profiles (RSTUSRPRF) command restores the basic parts of a user profile or a set of user profiles that were saved by the Save System (SAVSYS) command. The RSTUSRPRF command restores only the special authority granted in the Create User Profile (CRTUSRPRF) command; it does not restore the authority for the named objects (owned by other users). To do that, the Restore Authority (RSTAUT) command must be used after the profiles, libraries, and objects are restored. If all user profiles are being restored, authorization lists and authority holders are also restored. If all profiles are restored, all of the authorization lists and authority holders that existed at SAVSYS time are also restored.

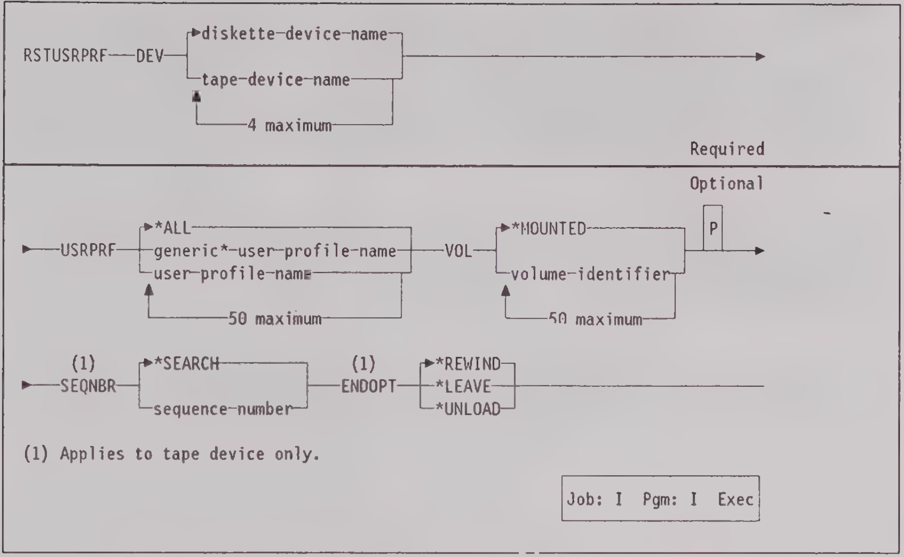
Before this command is specified, all other operations on the system must be stopped. This requires ending all subsystems through the End Subsystem (ENDSBS) or End System (ENDSYS) command or specifying this command when the operating system is started. The RSTUSRPRF command is normally used after the install operation but before the user libraries are restored. (The user profiles must be restored before any libraries or objects belonging to them can be restored.) After the libraries and their objects are restored, the authority for the objects is restored to the user profiles by the RSTAUT command. At the completion of the command either CPF3775 or CPC3705 is sent to QHST. More information on installing (restoring) the system is in the *Backup and Recovery Guide*.

The following situations may apply to user profiles being restored by the RSTUSRPRF command:

- If a user profile exists on the system, but is not on the media, the system profile remains.
- If a user profile exists on the media, but is not on the system, a new user profile is created.
- If the user profile exists on both the media and on the system, the media user profile is restored.
- If the user profile exists on the media and is being restored individually, the new user profile is created without its password or group connection.
- If the user profile exists on both the media and on the system, and is being restored individually, the media user profile is restored. However, the password and group connection on the system remains unchanged.
- If all user profiles are being restored, the passwords and group connections are also restored from the media.
- If user profiles exist on the system, there are no changes to the existing object authorities.

Note: This command ignores all file overrides that are currently in effect for the job.

Restriction: To use this command, you must have the save system authority in your user profile, specified by SPCAUT(*SAVSYS).



DEV Parameter

Specifies the name of the diskette device or tape device used to restore the user profiles. The device names must already be known on the system by a device description. For tape devices only, a maximum of four devices may be specified.

diskette-device-name: Specify the name of the diskette device used to restore the user profiles.

tape-device-name: Specify the name of one or more tape devices to be used to restore the user profiles. If you are using more than one tape device (up to a maximum of four), specify the names of the devices in the order you want them used. If using more than one tape volume, you may want to use more than one tape device so that one tape volume can rewind and/or unload while another tape device is processing.

USRPRF Parameter

Specifies the names of one or more user profiles to restore. To be restored, the user profiles must exist on the media from the Save System (SAVSYS) command.

*ALL: All of the user profiles saved by the SAVSYS command are restored.

generic-user-profile-name:* Specify one or more generic names of sets of user profiles to restore. (Add an asterisk after the last character in a generic name; for example ABC*. If an * is not included with the name, the system assumes that the name is a complete user profile name.)

user-profile-name: Specify one or more names of specific user profiles that are restored. Both generic names and specific names can be specified in the same command.

VOL Parameter

Specifies the volume identifiers of the tape volumes or diskette volumes from which the user profiles are restored. The volumes must be put on the devices in the same order as they were when the user profiles were saved by the Save System (SAVSYS) command. A maximum of 50 entries can be specified. (For more information on the VOL parameter, refer to Appendix A, "Expanded Parameter Descriptions.")

*MOUNTED: For diskette, the user profiles are restored from the volumes that are currently put on the device specified by the DEV parameter. For tape, the user profiles are restored from the volumes put on the tape devices, in the order that those devices were specified. This operation is normally performed immediately after an installation is done and the user profiles are on the same volume.

volume-identifier: Specify the identifiers of one or more volumes in the order they are put on the devices and used to restore the user profiles.

SEQNBR Parameter

Specifies the sequence number of the tape file used for the restore process.

*SEARCH: The volume put on is searched for a file containing the saved user profiles; when a match is found, the user profiles are restored. If the last operation on the device specifies ENDOPT(*LEAVE) (the tape is positioned at the location where the last operation ended), the file search starts with the first file beyond the current tape position. If ENDOPT(*LEAVE) was not used for the last operation (or if the tape was manually rewound since an ENDOPT(*LEAVE) operation), the search begins with the first file on the volume.

sequence-number: Specify the sequence number of the file used to restore user profiles.

ENDOPT Parameter
Specifies the positioning operation done automatically on the tape volume when the save/restore file is closed. In the case of a multiple-volume save/restore file, this parameter applies to the *last* reel only; all other reels are rewound and unloaded when the end of the tape is reached.

***REWIND:** The tape is rewound, but not unloaded, after the restore operation has completed.

***LEAVE:** The tape should be left in its current position after the restore operation has completed; it is not rewound or unloaded.

***UNLOAD:** The tape is rewound and unloaded after the restore operation has completed.

Examples

```
RSTUSRPRF DEV(TAP1) SEQNBR(*SEARCH) ENDOPT(*REWIND)
```

The saved version of all user profiles contained on the tape currently put on tape drive TAP1 is to be restored to the system. The file is searched for on the tape, and the tape is rewound on completion or ending of the restore.

```
RSTUSRPRF DEV(TAP1) USRPRF(USRA USRB USRC USER*)
```

The saved version of all the user profiles must exist on the tape put on tape drive TAP1. User profiles USRA, USRB, and USRC, along with all the user profiles whose names start with USER, are restored to the system.

```
RSTUSRPRF  DEV(DKT1)  VOL(*MOUNTED)
```

The saved version of all user profiles contained on the diskette volume currently put on is restored to the system.

RTVAUTLE (Retrieve Authorization List Entry) Command

The Retrieve Authorization List Entry (RTVAUTLE) command gets the authorities that a user has on the authorization list. It can be used with the Change Authorization List Entry command to change the user's rights to include new rights in addition to the existing rights for the user.

The authorization list name and user name must be specified. The variables for each of the authorities the user might have are returned blank if the user does not have the authority, and with the correct value for the Change Authorization List Entry (CHGAUTLE) command if the user does have the authority. The values are returned in the specified CL variables for the desired user.

Users with *AUTLMGT authority, who own the authorization list or have *ALLOBJ authority, can get authority for any user on the list. Other users can get their own authorities or the authority of *PUBLIC. The user must have *READ authority to the user profile the authorities are coming from.

The following values are returned if the user does have the authority specified by the keyword:

Keyword	Value
ADD	*ADD
ALL	*ALL
AUTLMGT	*AUTLMGT
CHANGE	*CHANGE
DELETE	*DLT
EXCLUDE	*EXCLUDE
OBJEXIST	*OBJEXIST
OBJMGT	*OBJMGT
OBJOPR	*OBJOPR
READ	*READ
UPDATE	*UPD
USE	*USE

The CL prompt for this command lists the minimum length for the variables next to the appropriate parameters you want to get. For character variables, a single number is shown.

```
RTVAUTLE──AUTL authorization-list-name─────────────────────────►

            ┌user-ID─┐ P
►──USER─────┤        ├─────────────────────────────────────────►
            └*PUBLIC─┘
                                                        Required
────────────────────────────────────────────────────────────────
                                                        Optional
►──ALL CL-variable-name──CHANGE CL-variable-name───────────────►

►──USE CL-variable-name──EXCLUDE CL-variable-name──────────────►

►──OBJEXIST CL-variable-name──OBJMGT CL-variable-name──────────►

►──OBJOPR CL-variable-name──READ CL-variable-name──────────────►

►──ADD CL-variable-name──UPDATE CL-variable-name───────────────►

►──DELETE CL-variable-name──AUTLMGT CL-variable-name────────────

                                              Pgm: B,I
```

AUTL Parameter
Specifies the name of the authorization list that the user's authorities come from.

USER Parameter
Specifies the name of the user whose information you want to get. If a variable is specified, it must be 10 characters in length and contain a user name or the value *PUBLIC.

user-ID: Specify the user ID of the user whose information you want to get.

***PUBLIC:** The information returned in the specified parameter is for the users who do not have any specific authority to the authorization list, and whose group does not have any specific authority to the authorization list.

ALL Parameter
Specifies the name of a 10-character CL variable that is used to return *ALL if the user has *ALL authority. Blanks are returned in the variable if the user does not have *ALL authority.

CHANGE Parameter
Specifies the name of a 10-character CL variable that is used to return *CHANGE if the user has *CHANGE authority. Blanks are returned in the variable if the user does not have *CHANGE authority.

USE Parameter
Specifies the name of a 10-character CL variable that is used to return *USE if the user has *USE authority. Blanks are returned in the variable if the user does not have *USE authority. The special value of *USE is returned in the CL variable.

EXCLUDE Parameter
Specifies the name of a 10-character CL variable that is used to return *EXCLUDE if the user has *EXCLUDE authority. Blanks are returned in the variable if the user does not have *EXCLUDE authority.

OBJEXIST Parameter
Specifies the name of a 10-character CL variable that is used to return *OBJEXIST if the user has *OBJEXIST authority. Blanks are returned in the variable if the user does not have *OBJEXIST authority.

OBJMGT Parameter
Specifies the name of a 10-character CL variable that is used to return *OBJMGT if the user has *OBJMGT authority. Blanks are returned in the variable if the user does not have *OBJMGT authority.

OBJOPR Parameter
Specifies the name of a 10-character CL variable that is used to return *OBJOPR if the user has *OBJOPR authority. Blanks are returned in the variable if the user does not have *OBJOPR authority.

READ Parameter
Specifies the name of a 10-character CL variable that is used to return *READ if the user has *READ authority. Blanks are returned in the variable if the user does not have *READ authority.

ADD Parameter
Specifies the name of a 10-character CL variable that is used to return *ADD if the user has *ADD authority. Blanks are returned in the variable if the user does not have *ADD authority. The special value of *ADD is returned in the CL variable.

UPDATE Parameter
Specifies the name of a 10-character CL variable that is used to return *UPD if the user has *UPD authority. Blanks are returned in the variable if the user does not have *UPD authority.

DELETE Parameter
Specifies the name of a 10-character CL variable that is used to return *DLT if the user has *DLT authority. Blanks are returned in the variable if the user does not have *DLT authority.

AUTLMGT Parameter
Specifies the name of a 10-character CL variable that is used to return *AUTLMGT if the user has *AUTLMGT authority. Blanks are returned in the variable if the user does not have *AUTLMGT authority.

Examples

The user has entered this command:

```
ADDAUTLE  AUTL(PAYROLL)  USER(TOM)  AUT(*OBJOPR *READ *UPD *AUTLMGT)
```

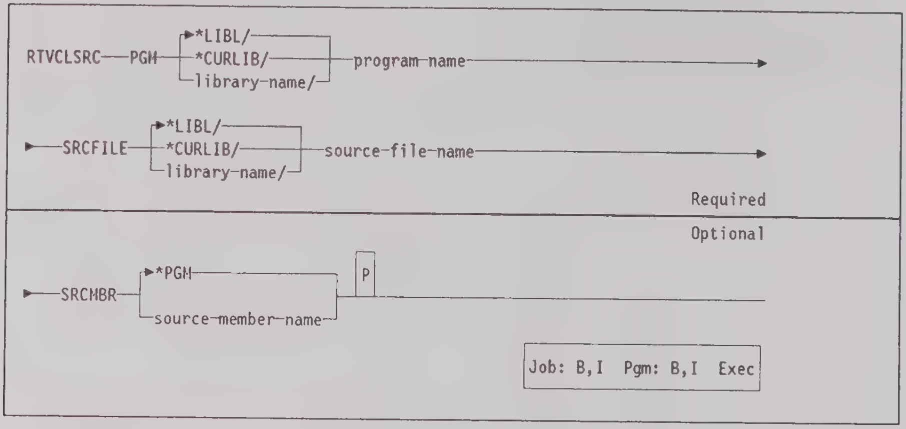
When user Smith calls a CL program containing the following:

```
DCL     &CHG         *CHAR 10
DCL     &ALL         *CHAR 10
DCL     &USE         *CHAR 10
DCL     &EXCL        *CHAR 10
DCL     &OBJOP       *CHAR 10
DCL     &READ        *CHAR 10
DCL     &ADD         *CHAR 10
DCL     &UPD         *CHAR 10
DCL     &DLT         *CHAR 10
DCL     &AUTLM       *CHAR 10

RTVAUTLE  AUTL(PAYROLL)  USER(TOM)  USE(&USE)  OBJOPR(&OBJOP)  AUTLMGT(&AUTLM)
```

This command gets the following rights from the authorization list PAYROLL for user TOM: *USE, *OBJOPR, *AUTLMGT. If TOM does not have the authority, blanks are returned.

RTVCLSRC (Retrieve CL Source) Command

The Retrieve CL Source (RTVCLSRC) command is used to retrieve the source statements from a CL program used to compile that program. These source statements are placed into a source file member, which can be used as input when recompiling the CL program.



PGM Parameter

Specifies the qualified name of the CL program whose source is being retrieved.

The possible library values are:

***LIBL:** The library list is used to locate the program.

***CURLIB:** The current library for the job is used to locate the program. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the program is located.

SRCFILE Parameter

Specifies the qualified name of the previously created data base source file into which the CL source statements are being written.

The possible library values are:

***LIBL:** The library list is used to locate the source file.

***CURLIB:** The current library for the job is used to locate the source file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the source file is located.

SRCMBR Parameter

Specifies the name of the data base source file member into which the CL source statements are being written. If not specified, the CL program name is assumed. If the member existed before running the command, it is cleared before any source statements are written into it. If the member did not exist, it is created.

***PGM:** The name of the CL program is used as the member name.

source-member-name: Specify the name of the source file member that contains the CL source statements.

Example

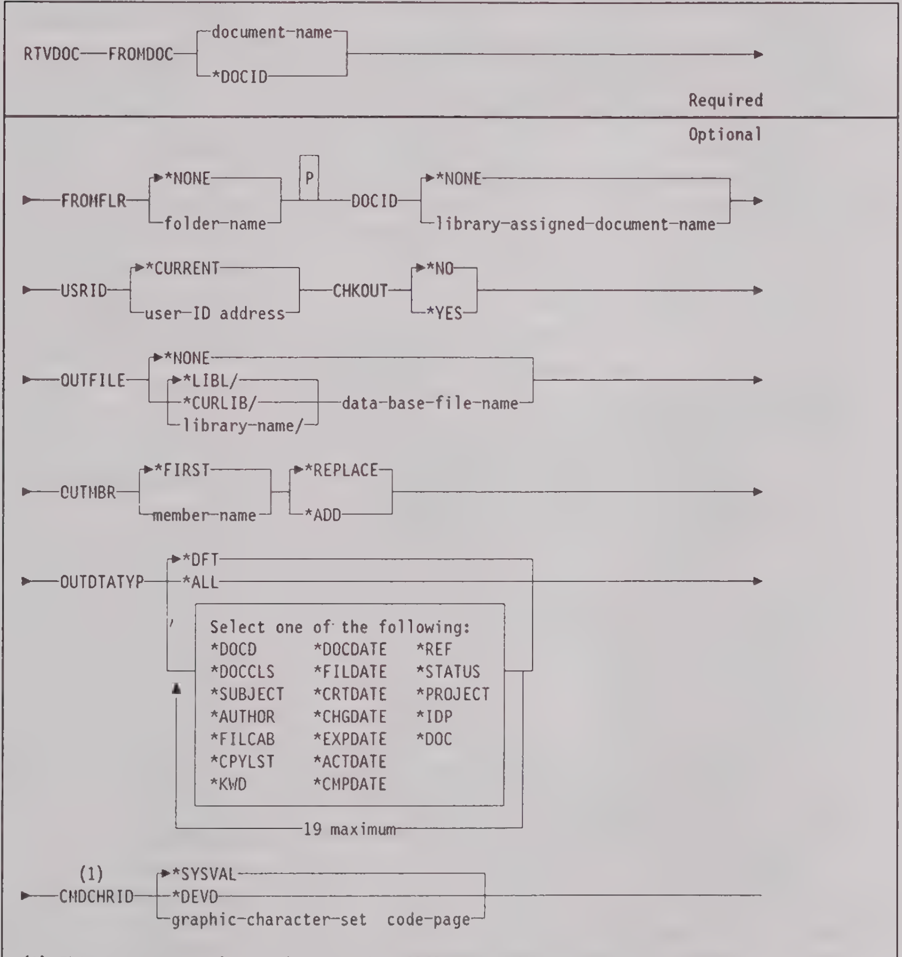
```
RTVCLSRC   PGM(JOHN1/TEXT1)  SRCFILE(JOHN2)  SRCMBR(JOHN3)
```

This command retrieves the source statements from the CL program named TEXT1 in library JOHN1. The retrieved source statements are placed into the file named JOHN2, and are named as member JOHN3.

RTVDOC (Retrieve Document) Command

The Retrieve Document (RTVDOC) command allows a user to copy information from a specific document to a data base file and allows the user to check out a document from the document library.

Restriction: If the current user of this command is working on behalf of another user, he must have the authority granted to him by the other user with the Grant User Permission (GRTUSRPMN) command to work on his behalf.



(1) This parameter value applies to the USRID parameter.

Job: B,I Pgm: B,I Exec

FROMDOC Parameter
Specifies the name of the document being retrieved.

document-name: Specify the user-assigned name of the document being retrieved.

***DOCID:** The document being retrieved is identified by the library-assigned document name specified in the DOCID parameter.

FROMFLR Parameter
Specifies the name of the folder that contains the document being retrieved. This is the name assigned to the folder when it is created. This parameter must be specified if a document name, FROMDOC parameter (FROMDOC(document-name)), is specified.

***NONE:** No folder name is specified when the document is identified by the DOCID parameter.

folder-name: Specify the name of the folder that contains the document retrieved.

DOCID Parameter

Specifies the library-assigned name of the document being retrieved. This is the name assigned to the document by the system when it is created. The library-assigned document names can be determined by using the Query Document Library (QRYDOCLIB) command or the library-assigned document name is returned in a completion message when using the File Document (FILDOC) command.

Library-assigned document names are 24 characters in length with the following format: YYYYMMDDHHMNSSHSSNSNSNSN where:

YYYY = year
MM = month
DD = day
HH = hour
MN = minute
SS = second
HS = hundredths of a second
SNSNSNSN = system name

__*NONE:__ No library-assigned document name is required when the document is identified by the FROMDOC parameter.

library-assigned-document-name: Specify the library-assigned document name sent. (More information on library-assigned document names is in the *Communications: Distribution Services Network Administrator's Guide*.)

USRID Parameter

Specifies the user ID and address that is associated with this request.

__*CURRENT:__ A document for the current user of this command is retrieved.

user-ID/address: Specify the user ID and address of the user that the document is retrieved for. The current user of this command must have authority to work on behalf of the user specified in this parameter.

CHKOUT Parameter

Specifies the document being retrieved can be replaced with new or changed data. If the document is read only, then specify CHKOUT(*NO). If the document being retrieved cannot be replaced, and CHKOUT(*YES) is specified, an error occurs.

Note: If a user is retrieving a document for checkout and a cancel request is entered, the document may or may not be checked out.

__*NO:__ The copy request only reads the data. Users requesting this function need only *USE authority to the document. The authority for the public is *READ authority.

*YES: The document data can be updated and replaced later. Users requesting this function must have *CHANGE authority. The document is unavailable to other users until a replace is done using the Replace Document Command (RPLDOC).

OUTFILE Parameter

Specifies the name of the data base file where the document data of the display is directed. If the OUTFILE does not exist, this command creates a data base file in the specified library. If the file is created by this function, the text saying that OUTFILE is created by RTVDOC is shown and the authority given to users with no specific authority is *EXCLUDE.

Note: The outfile format must be the same as the OSRTVD of the system file QSYS/QAOSIRTV.

__*NONE:__ The output is not directed to a data base file.

This file can be reused when other RTVDOC commands are used. Output from the file can start at the start of the file member or the data can be added to the file, depending on the OUTMBR parameter. The IBM-supplied data base file QAOSIRTV in library QSYS cannot be specified.

The possible library values are:

__*LIBL:__ If no library qualifier is specified, the library list (*LIBL) is used to locate the specified data base file.

*CURLIB: The current library is used to locate the specified data base file. If the specified current library entry exists in the library list, QGPL is used.

library-name: Specify the name of the library where the data base file is located.

data-base-file-name: Specify the name of the data base file (library name/database file name) to receive the output of the display. More information on defining the format of data base files (OUTFILES) is in the *Data Base Guide*.

OUTMBR Parameter

Specifies the name of the data base file member where the output is directed. If the data base file member does not exist by that name, one is created and stored in the specified library. If the member does exist, the system adds the records to it or clears the member and then adds the records to it.

__*FIRST:__ The first member in the file receives the output. If OUTMBR(*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified in the OUTFILE parameter.

member-name: Specify the named file member to receive the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

__*REPLACE:__ If a member exists, the system clears it and adds the new records.

***ADD:** If a member exists, the system adds the new records to the end of the existing records.

OUTDTATYP Parameter

Specifies what parts of information about the selected documents are written to the OUTFILE if an OUTFILE is specified.

***DFT:** The document information record is written to the outfile. This is the same as specifying OUTDTATYP(*DOCD) and OUTDTATYP(*DOC).

Record code – 105, Document description record
Record code – 110, Creation date
Record code – 115, Expiration date
Record code – 120, Document date
Record code – 125, File date
Record code – 130, Change date
Record code – 135, Action due date
Record code – 140, Completion date
Record code – 145, Author
Record code – 150, Copy list
Record code – 155, Document class
Record code – 160, File cabinet reference
Record code – 165, Subject
Record code – 170, Keyword
Record code – 175, Reference
Record code – 180, Status
Record code – 185, Project
Record code – 500, Interchange document profile data
Record code – 800, Document data

***ALL:** All information records about the document are written to the OUTFILE.

***DOCD:** The document information record is written to the OUTFILE.

***DOCCLS:** The document class record is written to the OUTFILE.

***FILCAB:** The filing cabinet reference code is written to the OUTFILE.

***SUBJECT:** The subject records are written to the OUTFILE.

***KWD:** The keyword records are written to the OUTFILE.

***AUTHOR:** The author records are written to the OUTFILE.

***CPYLST:** The copy list records are written to the OUTFILE.

***DOCDATE:** The document date record is written to the OUTFILE.

***FILDATE:** The file date record is written to the OUTFILE.

***CRTDATE:** The create date record is written to the OUTFILE.

***CHGDATE:** The date last changed record is written to the OUTFILE.

***EXPDATE:** The expiration date record is written to the OUTFILE.

***ACTDATE:** The action due date record is written to the OUTFILE.

***CMPDATE:** The completion date record is written to the OUTFILE.

***REF:** The reference record is written to the OUTFILE.

***STATUS:** The status record is written to the OUTFILE.

***PROJECT:** The project record is written to the OUTFILE.

***IDP:** The interchange document profile (IDP) is written to the OUTFILE.

***DOC:** The document data record is written to the OUTFILE.

CMDCHRID Parameter

Specifies the character identifier (graphic character set and code page) for the data being entered as command parameter values (applies to USRID parameter). The character identifier is related to the display device that was used to enter the command. (More information about CHRID processing is in the *Data Base Guide.*)

This value translates the USRID parameter to character set and code page set of '930 256'. The *Communications: Distribution Services Network Administrator's Guide* contains the character set and code page table for '930 256'.

***SYSVAL:** The system is to determine the graphic character set and code page values for the command parameter from QCHRID system value.

***DEVD:** The system is to determine the graphic character set and code page values for the command parameter from the display device description where this command was entered. This option is valid only when entered from an interactive job. If specified in an interactive CL program or a batch job, an error occurs.

graphic-character-set code-page: Specify the graphic character set and values that were used to create the command parameter. Both parts can be up to three characters in length and must be separated by at least one blank.

Examples

```
RTVDOC  FROMDOC(MYDOC)  FROMFLR(PERSONAL)  +
  USRID(*CURRENT)  OUTFILE(*CURLIB/MYFILE)  +
  OUTMBR(*FIRST)  MBROPT(*ADD)  OUTDTATYP(*ALL)
```

This command copies all information about document MYDOC located in folder PERSONAL for the current user of this command. CHECKOUT(*NO) is assumed; therefore, the document data can only be read. The output is directed to the data base file MYFILE in the user's current library and is added to the first member in that file.

```
RTVDOC  FROMDOC(SECOP)  FROMFLR(PERSONAL)  +
  USRID(MARY    SYSTEM1 )  CHKOUT(*YES)  +
  OUTFILE(MARLIB/SECFILE)  OUTMBR(*FIRST *ADD)
```

This command copies the default information (*DOCD and *DOC) about document SECOP located in folder PERSONAL for MARY. The document can be updated with new data and then replaced. The current user of this command must have the authority to work on behalf of MARY granted to him by Mary using the GRTUSRPMN command. The output is directed to the data base file SECFILE in the Mary's library MARLIB. The output is added to the first member of SECFILE.

RTVDTAARA (Retrieve Data Area) Command

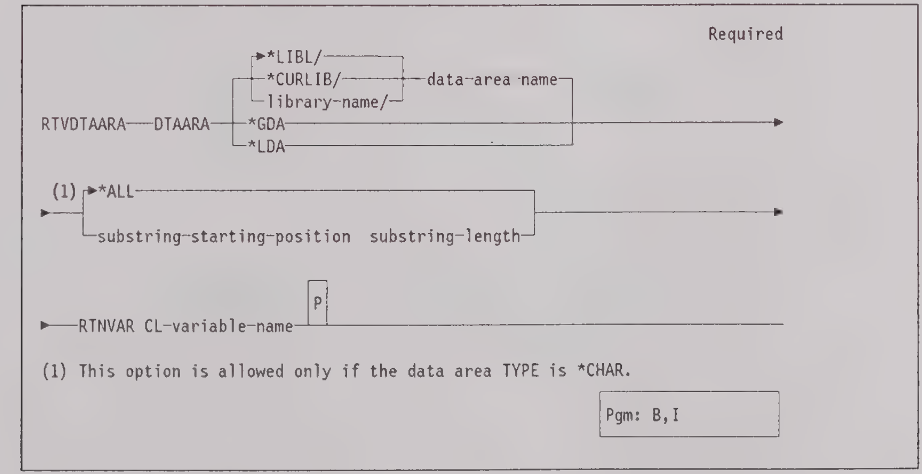
The Retrieve Data Area (RTVDTAARA) command is used in a CL program to retrieve all or part of a specified data area and copy the portion of the data area being retrieved into a CL variable within the CL program. RTVDTAARA does not retrieve any other attributes of the data area. The data area is not required to exist at the time the CL program is compiled.

If your job is a group job, the data area specified may be the group data area (*GDA). This data area is automatically associated with the group and is inaccessible from jobs outside the group. The length of this character data area is 512 bytes. The *Work Management Guide* has more information about group jobs.

A local data area (*LDA) is a character data area that is 1024 bytes in length, and is automatically associated with your job. Another job cannot access the local data area.

When the RTVDTAARA command is processed specifying other than a local data area or group data area, the data area is locked during the retrieval operation so that commands in other jobs cannot change or destroy the data area until the operation is complete. If the data area is shared with other jobs and is updated in steps involving more than one command in a job, the data area should be explicitly allocated to that job until all the steps have been performed. A data area other than a local data area or group data area can be explicitly allocated with the Allocate Object (ALCOBJ) command. (No allocation is necessary for a local data area or group data area.)

Restrictions: (1) This command is valid only in compiled CL programs. (2) To use this command, you must have *CHANGE authority for the data area and *USE authority for the library where the data area is located. No specific authority are necessary to retrieve the value of a local data area or group data area.



DTAARA Parameter
Specifies the qualified name of the data area whose value is retrieved.

The possible library values are:

***LIBL:** The library list is used to locate the data area.

***CURLIB:** The current library for the job is used to locate the data area. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the data area is located.

data-area-name: Specify the name of the data area whose value is retrieved.

***GDA:** Specifies that the value of the group data area is being retrieved. This value is only valid if this job is a group job.

***LDA:** Specifies that the value of the local data area is being retrieved.

The substring retrieval option, valid for character data areas only, specifies the starting position and the length of the portion of the data area that is being retrieved into the CL character variable.

ALL: Specifies that the entire data area is retrieved.

substring-starting-position: Specifies the starting position of the data area being retrieved.

substring-length: Specifies the length of the data area substring being retrieved.

Note that it is not possible to retrieve data outside the bounds of the data area. The combination of starting position and length must always specify positions within the data area.

RTNVAR Parameter

Specifies the name of the CL program variable that receives the contents of the data area being returned.

No type conversion is performed by the RTVDTAARA command:

- If RTNVAR is declared as TYPE(*DEC), the data area retrieved must be TYPE(*DEC).
- If RTNVAR is declared as TYPE(*CHAR), the data area retrieved must be either TYPE(*CHAR) or TYPE(*LGL).
- If RTNVAR is declared as TYPE(*LGL), the data area retrieved must be either TYPE(*LGL) or TYPE(*CHAR) with a value of either '0' or '1'.

If a retrieved character string is shorter than the length of the variable specified by the RTNVAR parameter, the value is padded to the right with blanks. The retrieved string length must be less than or equal to the CL variable length.

When decimal data areas are retrieved, the decimals are aligned. The value of the integer portion of the data area must be able to be placed into the integer portions of the CL variable. Fractional data is truncated if the fraction contains more digits than the CL variable.

Character Data Area Example

Assume data area DA1 has been created by the following command:

```
CRTDTAARA   DTAARA(DA1)  TYPE(*CHAR)  LEN(3)  VALUE(ABC)
```

and variable &CLVAR1 has been declared as follows:

```
DCL   VAR(&CLVAR1)  TYPE(*CHAR) LEN(5)  VALUE(VWXYZ)
```

Running the command:

```
RTVDTAARA   DTAARA(DA1)  RTNVAR(&CLVAR1)
```

results in the following:

```
&CLVAR1 = 'ABC  '
```

Running the command:

```
RTVDTAARA   DTAARA(DA1 (2 1))  RTNVAR(&CLVAR1)
```

results in the following:

```
&CLVAR1 = 'B    '
```

Decimal Data Area Example

Assume data area DA2 has been created with the following attributes:

```
CRTDTAARA   DTAARA(DA2)  TYPE(*DEC)  LEN(5 2)  VALUE(12.39)
```

and variable &CLVAR2 has been declared as follows:

```
DCL   VAR(&CLVAR2)  TYPE(*DEC)  LEN(5 1)  VALUE(4567.8)
```

Running this command:

```
RTVDTAARA   DTAARA(MYLIB/&DTAARA)  RTNVAR(&CLVAR2)
```

results in the following:

```
CLVAR2 = 0012.3
```

Note that fractional digits are truncated instead of rounded.

RTVGRPA (Retrieve Group Attributes) Command

The Retrieve Group Attributes (RTVGRPA) command retrieves information about the group where the job that issued the RTVGRPA command belongs. The following attributes can be retrieved:

- The group job name of the job calling the RTVGRPA command
- A list containing information about all active jobs in the group
- A count of the number of active jobs in the group
- The name of the group message queue
- The library that the group message queue resides in
- The group job name and job number of the previously active job in the group
- A control code indicating why the currently active job in the group gained control

The prompt for this command lists the minimum length for retrieved variables next to the appropriate parameters. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length, and the second number indicates the minimum number of decimal positions.

```
                                                                Optional
RTVGRPA──GRPJOB CL-variable──GRPJOBL CL-variable─────────────────────────►

                             ┌─┐
                             │P│
►──GRPJOBCNT CL-variable─────┴─┴──MSGQ CL-variable──MSGQLIB CL-variable──────►

►──PRVGRPJOB CL-variable──CTLCDE CL-variable──────────────────────────────────

                                                           Pgm: I
```

GRPJOB Parameter
Specifies the name of the CL variable that receives the group job name of the job. The variable must be a character variable with a minimum length of 10 characters. (If the group job name has fewer characters than the variable allows, the value is padded on the right with blanks.)

Note: This is the only parameter of the RTVGRPA command that is meaningful if the job that issued the RTVGRPA command is not a group job. When this occurs, the variable is set to the special value of *NONE; any other parameters specified in the command are filled with zeros or blanks (numeric variables with zeros, character variables with blanks). If this parameter is not specified and the job issuing this command is not a group job, an exception is signaled.

GRPJOBL Parameter
Specifies the name of the CL variable that receives the list of jobs in the group. Each entry contains the job's group job name, the job's job number, and the 50 characters of descriptive text for the group job.

The maximum number of entries in the variable is 16. The entries are ordered by most recently active job (the group job that is issuing the RTVGRPA command is always first in the variable). The variable must be a character variable with a minimum length of 1056 characters. (If the group job list has fewer characters than the variable allows, the value is padded on the right with blanks.)

The group job list has the the following format:

Entry-1

Group-job-name	CHAR(10)
Job-number	CHAR(6)
Group-job-text	CHAR(50)

Entry-2

.
.
.

Entry-16

Group-job-name	CHAR(10)
Job-number	CHAR(6)
Group-job-text	CHAR(50)

This list contains all of the active jobs in the group. Jobs in the group that have not completely ended (jobs that have been canceled) are not listed.

GRPJOBCNT Parameter
Specifies the CL variable that receives the count of active jobs in the group. The CL variable must be a three-position decimal variable with no decimal positions.

The CL variable contains the number of nonblank entries in the group job list. The count includes all of the active jobs in the group. Jobs that have not completely ended (jobs that have been canceled) are not counted.

MSGQ Parameter

Specifies the name of the CL variable that receives the group message queue name. This variable must be a character variable with a minimum length of 10 characters. (If the message queue name has fewer characters than the variable allows, the value is padded on the right with blanks.)

If there is no message queue associated with the group, the CL variable is set to the special value of *NONE.

MSGQLIB Parameter

Specifies the name of the CL variable that receives the name of the library that contains the group message queue. This variable must be a character variable with a minimum length of 10 characters. (If the library name has fewer characters than the variable allows, the value is padded on the right with blanks). If there is no message queue associated with the group, the CL variable is set to blanks.

Refer to the Change Group Attributes (CHGGRPA) command for more information on the use of the group message queue.

PRVGRPJOB Parameter

Specifies the name of the CL variable that receives the group job name and job number of the previously active job in the group. The variable must be a character variable with a minimum length of 16 characters. (If the group job name has fewer characters than the variable allows, the value is padded on the right with blanks.)

If there is no previously active job in the group (no Transfer to Group Job [TFRGRPJOB] command is issued), the group job name portion of the CL variable is set to the special value of *NONE, and the job number portion of the CL variable is set to blanks. The CL variable is returned in the following format:

```
Group-job-name     CHAR(10)
Job-number   CHAR(6)
```

Even though a group job name and job number are returned, it is not necessarily true that the group job still exists. There are cases where the previously active job in the group has ended (which caused the currently active job to become active).

CTLCDE Parameter

Specifies the name of the CL variable that receives information about why the active job in the group has gained control. The CL variable must be a three-position decimal variable with no decimal positions. The following control codes (and their meanings) are possible:

0	There was no previously active job (no TFRGRPJOB commands have been run for this group).
10	The previously active job selected this job as the job to be transferred to on the TFRGRPJOB command.
20	The previously active job's first group program ended normally, and this job was the most recently active job in the group.
30	The previously active job was ended by the End Group Job (ENDGRPJOB) command, and this job was selected to gain control (the RSMGRPJOB parameter specified this group job).
40	The previously active job was ended by the End Group Job (ENDGRPJOB) command and selected a job other than this job to gain control (which was ended before it could be resumed). Since this job was the most recently active job in the group, control has been passed to it.
50	The previously active job was ended by the End Group Job (ENDGRPJOB) command, and this job was the most recently active job in the group (the RSMGRPJOB parameter specified *PRV).
60	The previously active job's first group program ended abnormally, and this job was the most recently active job in the group.
70	The previously active job was ended by the End Job (ENDJOB) command, and this job was the most recently active job in the group.

Examples

Assume jobs 0001/QUSER/WORKST01 and 0002/QUSER/WORKST01 are group jobs with group job names GROUPJ1 and GROUPJ2 (respectively). Also assume that message queue QGPL/GROUPMSGQ is associated with the group. If group job GROUPJ1 has just issued the TFRGRPJOB command to transfer to group job GROUPJ2, and GROUPJ2 called the following CL program:

PGM Example

```
DCL VAR(&GRPJOBN)      TYPE(*CHAR)     LEN(10)
DCL VAR(&GRPJOBL)      TYPE(*CHAR)     LEN(1056)
DCL VAR(&GRPCOUNT)     TYPE(*DEC)      LEN(3 0)
DCL VAR(&MSGQNAME)     TYPE(*CHAR)     LEN(10)
DCL VAR(&MSGQLIB)      TYPE(*CHAR)     LEN(10)
DCL VAR(&PRVJOB)       TYPE(*CHAR)     LEN(16)
DCL VAR(&CTLCODE)      TYPE(*DEC)      LEN(3 0)

  RTVGRPA  GRPJOB(&GRPJOBN)  GRPJOBL(&GRPJOBL)
  GRPJOBCNT(&GRPCOUNT)  MSGQ(&MSGQNAME)
  MSGQLIB(&MSGQLIB)  PRVGRPJOB(&PRVJOB)
  CTLCDE(&CTLCODE)
```

The contents of the CL variables returned are as follows:

```
&GRPJOBN: GROUPJ2
&GRPJOBL:

GROUPJ2     0002     50 characters of text for this group job...
GROUPJ1     0001     50 characters of text for this group job...
```

Fourteen more entries, full of blanks

```
&GRPCOUNT:     002
&MSGQNAME:     GROUPMSGQ
&MSGQLIB:      QGPL
&PRVJOB:       GROUPJ1     0001
&CTLCODE:      010
```

RTVJOBA (Retrieve Job Attributes) Command

The Retrieve Job Attributes (RTVJOBA) command is used in a CL program to retrieve the values of one or more job attributes and place the values into the specified CL variable. The attributes are retrieved for the job in which this command is used. The following attributes can be retrieved:

- The job name (JOB), user profile name (USER), and job number (NBR)
- The message logging level (LOGLVL), severity level (LOGSEV), and text level (LOGTYPE)
- The CL program logging flag (LOGCLPGM) value
- The inquiry message handling (INQMSGRPY) value
- The output queue (OUTQ) being used for spooled output and the library (OUTQLIB) containing the output queue
- The accounting code (ACGCDE) for the job
- The date when the job was started (DATE)
- The values of the eight job switches (SWS) associated with the job
- The job type (TYPE), which can be batch or interactive
- The return code (RTNCDE) set by the last RPG, COBOL, DFU, or conversion reformat utility program that has completed processing in the job
- The cancellation status (CNLSTS), which indicates whether a controlled cancel operation affecting the job is in progress
- The running (or processing) priority (RUNPTY) and the time slice (TIMESLICE) for the job's routing steps
- The value of the main storage purge flag (PURGE)
- The default wait time (DFTWAIT) for routing step processing
- The user portion of the job's library list (USRLIBL)
- The submitter specified job completion message queue (SBMMSGQ) and the library that contains it (SBMMSGQLIB)
- The current value of the print text (PRTTXT) job attribute
- The Distributed Data Management conversation setting (DDMCNV)
- The job's break message handling mode (BRKMSG)
- The date format (DATFMT) and date separator character (DATSEP) for the job.
- The name of the current library for the job (CURLIB).

The CL prompt for this command lists the minimum length for retrieved variables next to the appropriate parameters. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length and the second number indicates the minimum number of decimal positions.

Restriction: This command is valid only within a CL program.

```
                                                                    Optional
RTVJOBA──JOB CL-variable-name──USER CL-variable-name────────────────────────►

                          ┌─┐
                          │P│
►──NBR CL-variable-name───┴─┴──LOGLVL CL-variable-name──────────────────────►

►──LOGSEV CL-variable-name──LOGTYPE CL-variable-name────────────────────────►

►──LOGCLPGM CL-variable-name──INQMSGRPY CL-variable-name────────────────────►

►──OUTQ CL-variable-name──OUTQLIB CL-variable-name──────────────────────────►

►──ACGCDE CL-variable-name──DATE CL-variable-name──SWS CL-variable-name─────►

►──TYPE CL-variable-name──RTNCDE CL-variable-name───────────────────────────►

►──CNLSTS CL-variable-name──RUNPTY CL-variable-name─────────────────────────►

►──TIMESLICE CL-variable-name──PURGE CL-variable-name───────────────────────►

►──DFTWAIT CL-variable-name──USRLIBL CL-variable-name───────────────────────►

►──SBMMSGQ CL-variable-name──SBMMSGQLIB CL-variable-name────────────────────►

►──PRTTXT CL-variable-name──DDMCNV CL-variable-name─────────────────────────►

►──BRKMSG CL-variable-name──DATFMT CL-variable-name─────────────────────────►

►──DATSEP CL-variable-name──CURLIB CL-variable-name─────────────────────────►

►──PRTDEV CL-variable-name──SYSLIBL CL-variable-name────────────────────────

                                                        ┌───────────────┐
                                                        │Pgm: B,I       │
                                                        └───────────────┘
```

JOB Parameter
Specifies, if the job name is returned, the name of the CL variable that receives the name of the job. The variable must be a character variable with a minimum length of 10 characters. (If the job name has fewer characters than the variable allows, the value is padded on the right with blanks.)

USER Parameter
Specifies, if the user name is returned, the name of the CL variable that receives the name of the user profile associated with the job when the job was started. The user name is the second part of the qualified job name. The variable must be a character variable with a minimum length of 10 characters. (If the user name has fewer characters than the variable allows, the value is padded on the right with blanks.)

NBR Parameter
Specifies, if the job number is being returned, the name of the CL variable that receives the unique 6-character number assigned to the job by the system. The job number is the first part of the qualified job name.

LOGLVL Parameter
Specifies the name of the CL variable that receives the 1-character value, in the range from 0 to 4 that is the message logging level being used to determine the type of messages logged in the job log. The variable must be a character variable with a minimum length of 1 character.

LOGSEV Parameter
Specifies the name of the CL variable that receives the 2-digit value, in the range of 0 through 99, which is the minimum severity level a message must have before it is logged in the job log. The variable must be a 2-digit decimal variable specified with 0 decimal positions.

LOGTYPE Parameter
Specifies the name of the CL variable that receives the special value which indicates the level of text that will appear for any message that is written to the job log. The variable must be a character variable with a minimum length of 10 characters. (If the special value has fewer characters than the variable allows, the value is padded on the right with blanks.)

LOGCLPGM Parameter
Specifies the name of the CL variable that receives the special value which indicates if processed commands in a CL program are being logged in the job log. (Note that this logging flag only has meaning for commands processed in CL programs which were created with LOG(*JOB) specified on the Create CL Program (CRTCLPGM) command. For additional information on the job's CL program logging flag, refer to the LOGCLPGM

parameter description in the Change Job (CHGJOB) command.) The variable must be a character variable with a minimum length of 10 characters. (If the special value has fewer characters than the variable allows, the value is padded on the right with blanks.)

INQMSGRPY Parameter
Specifies the name of the CL variable that receives the special value which indicates how inquiry messages are being handled by the job. (For additional information on the value returned in the variable, refer to the INQMSGRPY parameter description in the Change Job (CHGJOB) command.) The variable must be a character variable with a minimum length of 10 characters. (If the special value has fewer characters than the variable allows, the value is padded on the right with blanks.)

OUTQ Parameter
Specifies the name of the CL variable that receives the name of the output queue that is being used by the job for spooled output. The variable must be a character variable with a minimum length of 10 characters. (If the output queue name has fewer characters than the variable allows, the value is padded on the right with blanks.)

OUTQLIB Parameter
Specifies the name of the CL variable that receives the name of the library containing the output queue that is being used by the job for spooled output. The library name is the second part of the qualified output queue name (*output-queue.library-name*). The variable must be a character variable with a minimum length of 10 characters. (If the output queue library name has fewer characters than the variable allows, the value is padded on the right with blanks.)

ACGCDE Parameter
Specifies the name of the CL variable that receives the accounting code for the job. The variable must be a character variable with a minimum length of 15 characters. (If the accounting code has fewer characters than the variable allows, the value is padded on the right with blanks.)

DATE Parameter
Specifies, if the job date is to be returned, the name of the CL variable that receives the date assigned to the job by the system when the job was started. The variable must be a character variable with a minimum length of 6 characters. (The job date is returned in the system date format specified by the system value QDATFMT.)

SWS Parameter
Specifies, if the job switches are to be returned, the name of the CL variable that receives the value of the eight job switches used by the job. The job switches are retrieved as a single 8-character value with each of the characters specifying a 1 or 0 as the value of the associated switch. The CL variable must be a character variable with a minimum length of 8 characters.

TYPE Parameter
Specifies, if the type of job environment is being returned, the name of the CL variable that receives the 1-character value representing the environment of the job. A character value of 0 indicates the job is running as a batch job, and a 1 indicates an interactive job. The variable must be a character variable with a minimum length of 1 character.

RTNCDE Parameter
Specifies, if the completion status of an RPG, COBOL, DFU, or conversion reformat utility program is being returned, the name of the CL variable that receives the 5-digit decimal return code. The return code is set by these programs before they return to the programs that called them. The return code indicates the completion status of the last program (of these types) that has completed processing within the job, as follows:

0	Normal return (RPG, COBOL, DFU, or Conversion Reformat Utility)
1	LR (last record) indicator on (RPG)
2	Error – no halt indicator set (RPG, COBOL, DFU, or Conversion Reformat Utility)
3	Halt indicator set on (one of the RPG indicators H1 through H9)

The CL variable must be a five-position decimal variable with no decimal positions.

CNLSTS Parameter
Specifies, if checking for a controlled cancel operation, the name of the CL variable that receives the cancellation status. The single-character value indicates whether a controlled cancel that affects the job is currently being performed. A value of 1 indicates that the system, the subsystem in which the job is running, or the job itself is being canceled; a 0 indicates no controlled cancel is being performed. The CL variable must be a character variable with a minimum length of 1 character.

RUNPTY Parameter
Specifies the name of the CL variable that receives the 2-digit value, in the range of 1 through 99, which is the run (or processing) priority for routing steps that are part of the job. (For additional information on run priority, refer to the RUNPTY parameter description in the CHGJOB (Change Job) command.) The variable must be a 2-digit decimal variable specified with 0 decimal positions.

TIMESLICE Parameter
Specifies the name of the CL variable that receives the 7-digit value, in the range of 1 through 9999999, which is the maximum number of milliseconds that a job's routing step can run when it is given processing time. (For additional information on time slice, refer to the TIMESLICE parameter description under the CHGJOB (Change Job) command.) The variable must be a 7-digit decimal variable specified with 0 decimal positions.

PURGE Parameter
Specifies the name of the CL variable that receives the special value which indicates whether the job is marked as eligible to move from main storage at the end of a time slice or a long wait condition. (For additional information on job purging, refer to the PURGE parameter description under the CHGJOB (Change Job) command.) The variable must be a character variable with a minimum length of 10 characters. (If the special value has fewer characters than the variable allows, the value is padded on the right with blanks.)

DFTWAIT Parameter
Specifies the name of the CL variable that receives the 7-digit value, in the range of 1 through 9999999 (or -1 if the value is set to *NOMAX), which is the default for the maximum number of seconds that the system waits for a machine instruction to be completely processed. (For additional information on wait-time, refer to the DFTWAIT parameter description under the CHGJOB (Change Job) command.) The variable must be a 7-digit decimal variable specified with 0 decimal positions.

USRLIBL Parameter
Specifies the name of the CL variable that receives the user portion of the job's library list. The variable must be a character variable with a maximum of 275. Each library name returned is left-justified in an 11-character field and padded on the right with blanks.

```
DCL &LIBL *CHAR 275
DCL &CHGLIBL *CHAR 285
RTVJOBA USRLIBL(&LIBL)       /*  save library list   */
   •
   •
   •
CHGLIBL LIBL(MYLIB QGPL)     /*  temporarily change  */
                             /*  library list        */
   •
   •
   •
CHGVAR &CHGLIBL ('CHGLIBL (' *CAT &LIBL *TCAT ')')
                             /*  build               */
                             /*  command string      */
CALL QCMDEXC (&CHGLIBL 285)  /*  restore library     */
                             /*  list                */
```

The above commands retrieve the user portion of the library list so that later it can be restored from its temporary state, where only MYLIB and QGPL were in the user portion of the library list, to its original state. If there are no libraries on the user portion of the library list, then blanks are returned in the variable.

PRTTXT Parameter
Specifies the name of the CL variable that receives the print text for the job. The variable must be a character variable with a minimum length of 30 characters.

SBMMSGQ Parameter
Specifies the name of the CL variable that receives the name of a message queue. If this parameter is coded for a RTVJOBA command in a CL program which is part of a batch job, then the value returned is the value associated with the MSGQ parameter on the JOB or SBMJOB command which caused the batch job to start running. If *NONE was specified for the MSGQ parameter of the JOB or SBMJOB commands, or the CL program containing the RTVJOBA command is not part of a batch job, then *NONE is returned to the CL variable coded for this parameter. The variable must be a character variable with a minimum length of 10 characters. (If the message queue name has fewer characters than the variable allows, the value is padded on the right with blanks.)

SBMMSGQLIB Parameter
Specifies the name of the CL variable that receives the name of the library containing the message queue described above. (See the SBMMSGQ parameter description above.) The library name is the second part of the qualified message queue name. If *NONE is returned for the SBMMSGQ parameter, it is also returned for this parameter. The variable must be a character variable with a minimum length of 10 characters. (If the message queue library name has fewer characters than the variable allows, the value is padded on the right with blanks.)

DDMCNV Parameter
Specifies the name of a CL variable that receives the special value which indicates the action taken for distributed data management conversations on the job. The variable must be a character variable with a minimum length of 5 characters. If the special value has fewer characters than the variable allows, the value is filled on the right with blanks.

BRKMSG Parameter
Specifies the name of a CL variable that receives the special value which indicates the mode for break message handling that is in effect for the job. The variable must be a character variable with a minimum length of seven characters. If the special

value has fewer characters than the variable allows, the value is filled on the right with blanks.

DATFMT Parameter
Specifies the name of a CL variable that receives the special value being used as the date format for the job. The variable must be a character variable with a minimum length of four characters. If the special value has fewer than four characters, the value is filled on the right with blanks.

DATSEP Parameter
Specifies the name of a CL variable that receives the character being used as the date separator character for the job. The variable must be a character variable with a minimum length of one character.

CURLIB Parameter
Specifies the name of a CL variable that receives the name of the current library for the job. The variable must be a character variable with a minimum length of four characters. If the special value has fewer than four characters, the value is filled on the right with blanks.

PRTDEV Parameter
Specifies the name of the CL variable that receives the name of the printer device. The variable must be a character variable with a minimum length of 10 characters. If the printer device has fewer characters than the variable is padded on the right with blanks.

SYSLIBL Parameter
Specifies the name of the CL variable that receives the system portion of the job's library list. The variable must be a character variable with a maximum of 165. Each library name returned is left-justified in an 11-character field and padded on the right with blanks.

Examples

```
RTVJOBA   NBR(&JOBNBR)  DATE(&JOBDATE)
```

This command retrieves the job number and job date attributes from the job that this command is in. The 6-digit job number is to be copied into the CL variable &JOBNBR. The job date is to be copied into the CL variable &JOBDATE; both variables must be character variables with a length of six. The format of the date is determined by the contents of the system value QDATFMT, which controls the system date format.

```
DCL &LIBL *CHAR 275
DCL &CHGLIBL *CHAR 285
RTVJOBA USRLIBL(&LIBL                   /*  save library list   */
•
•
•
CHGLIBL LIBL(MYLIB QGPL)                /*  temporarily change  */
•                                       /*  library list        */
•
•
CHGVAR &CHGLIBL ('CHGLIBL (' *CAT &LIBL *TCAT ')')   /*  build  */
                                            /*  command string  */

CALL QCMDEXC (&CHGLIBL 285)           /*  restore library list  */
```

The above command retrieves the user portion of the library list so that later it can be restored from its temporary state, where only MYLIB and QGPL were in the user portion of the library list, to its original state.

If there are no libraries on the user portion of the library list, then blanks are returned in the variable. If a library on the library list has been deleted, the value '*DELETED' is put in the variable position for that name.

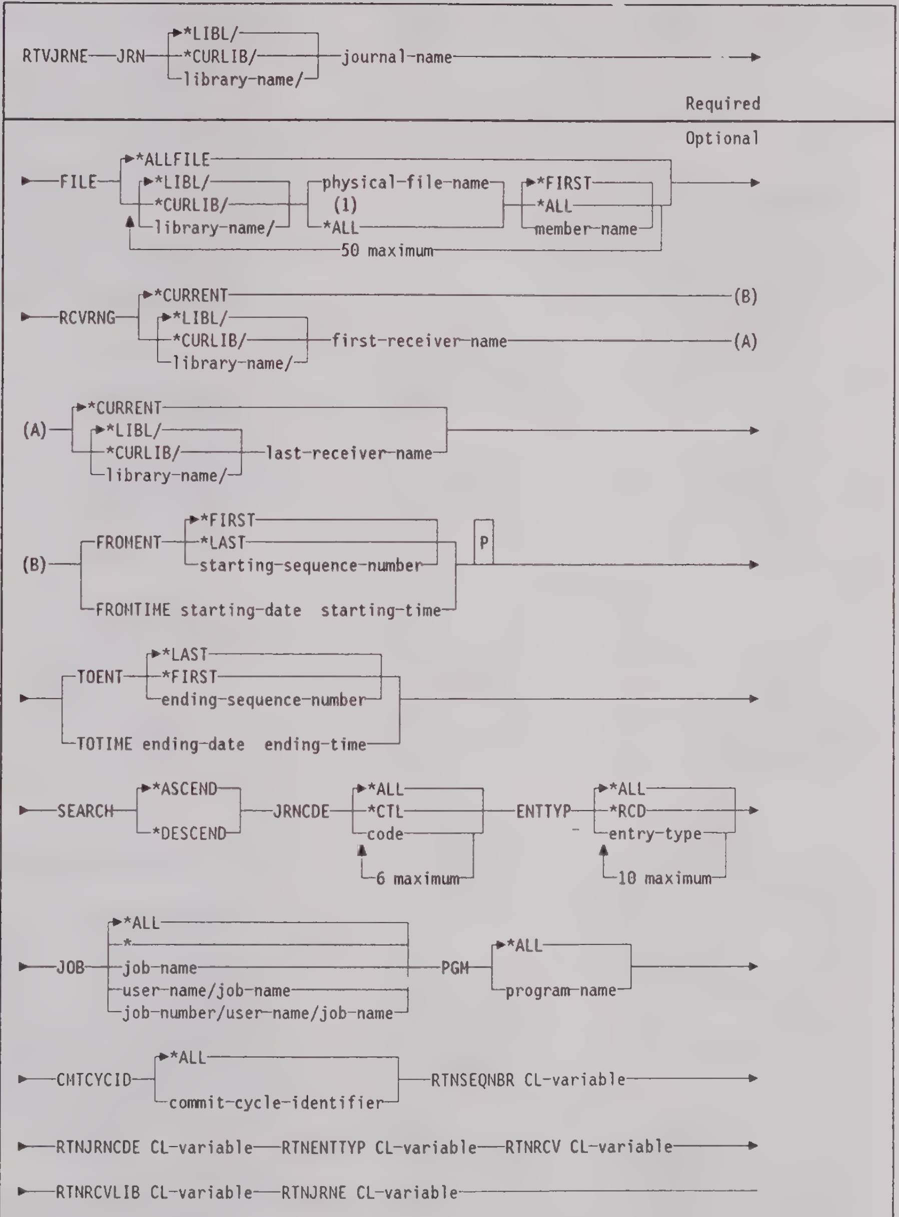
RTVJRNE (Retrieve Journal Entry) Command

The Retrieve Journal Entry (RTVJRNE) command supplies you with a means of getting a particular journal entry and placing the results in CL variables. The CL variables would contain information, such as the sequence number of the retrieved entry, and could be useful in automating certain types of recovery functions. The search for a particular journal entry can be restricted to a particular file, to a particular range of journal receivers, to a range of journal entries, to a journal code, to an entry type, to a particular job, to a particular program, or to a particular commit cycle identifier. Multiple limitation criteria can be specified. If more than one journal entry satisfies the search value specified, the first occurrence of a journal entry satisfying all of the specified search value is returned. If there is no journal entry satisfying the search value specified, the command ends with an escape message, and the return CL variables (RTNSEQNBR, RTNJRNCDE, RTNENTTYP, RTNRCV, RTNRCVLIB and RTNJRNE) remain the same.

The order of the search through the journal entries can be ascending or descending. The search order is determined by the value specified on the SEARCH parameter. The value for the FROM parameter must come before the value specified for the TO parameter in the search order specified.

The CL prompt for this command lists the minimum length for retrieved variables next to the correct parameters. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length, and the second number indicates the minimum number of decimal positions.

Restriction: If the sequence number is reset in the range of receivers specified, the first occurrence of FROMENT or TOENT is used, if they are specified.



(1) If *ALL is specified instead of file-name, the format must be library-name/*ALL.

Pgm: B,I

JRN Parameter

Specifies the qualified name of the journal from which the journal entry is retrieved (library-name/journal-name).

The possible library values are:

***LIBL:** The library list is used to locate the journal.

***CURLIB:** The current library for the job is used to locate the journal. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the journal is located.

FILE Parameter

Identifies the file and member for which a journal entry is retrieved.

The FILE parameter also specifies the name of the member in the file for which a journal entry is retrieved.

***ALLFILE:** The search for the entry to be retrieved is not limited by a particular file name.

The possible library values are:

***LIBL:** The library list is used to locate the physical file.

***CURLIB:** The current library for the job is used to locate the physical file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the physical file is located.

physical-file-name: Specify the name of the physical data base file for which a journal entry is retrieved (library-name/physical-file-name).

***ALL:** The search for the entry to be retrieved is for all files in the library that are currently being journaled. When FILE(*ALL) is specified, the file library name cannot be *LIBL.

***FIRST:** An entry for the first member in the file is retrieved.

***ALL:** Journal entries for all members in the file are considered for retrieval.

member-name: Specify the name of the member for which an entry is retrieved.

RCVRNG Parameter

Specifies the first (beginning) and last (ending) journal receivers used in the search for a journal entry to get. The system starts the search with the first journal receiver (as specified by the first value) and proceeds through the receiver chain until the last receiver (as specified by the last value) is processed. If dual receivers are used at any time, the first of the receivers is used when chaining through the receivers. (The Work with Journal Attributes [WRKJRNA] command can be used to display the order of the receivers in the receiver chain.) If any problem is found in the receiver chain before the search operation begins (such as damaged or off-line receivers), the system attempts to use the second of the dual receivers. If the second of the receivers is damaged or off-line, or if a problem is found during the operation, the operation ends.

If SEARCH(*ASCEND) is specified, journal receivers must be specified in the order of oldest to newest. If SEARCH(*DESCEND) is specified, journal receivers must be specified in the order of newest to oldest.

First Parameter Value:

***CURRENT:** The journal entries in the currently attached receiver are searched for the entry to get.

The possible library values are:

***LIBL:** The library list is used to locate the journal receiver.

***CURLIB:** The current library for the job is used to locate the journal receiver. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the journal receiver is located.

first-receiver-name: Specify the name of the first journal receiver containing journal entries to be searched. (library-name/first-receiver-name).

Second Parameter Value:

***CURRENT:** The search for a journal entry continues for all journal receivers (in the chain that began with the receiver specified by the first parameter value) through the currently attached journal receiver.

The possible library values are:

***LIBL:** The library list is used to locate the journal receiver.

***CURLIB:** The current library for the job is used to locate the journal receiver. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the journal receiver is located.

last-receiver-name: Specify the name of the last journal receiver containing journal entries to be searched (library-name/last-receiver-name). If the end of the receiver chain is reached before a receiver of this name, an exception is signaled and no journal entry is retrieved.

Note: If the maximum number of receivers in the range is larger than 256, an exception is signaled and no journal entry is retrieved.

FROMENT Parameter

Specifies the first journal entry considered for retrieval.

***FIRST:** The first journal entry in the journal receiver range specified is the first entry considered for retrieval. If SEARCH(*DESCEND) is specified, FROMENT(*FIRST) is valid only if TOENT(*FIRST) is also specified.

***LAST:** The last journal entry in the journal receiver range specified is the first entry considered for retrieval. If SEARCH(*ASCEND) is specified, FROMENT(*LAST) is valid only if TOENT(*LAST) is also specified.

starting-sequence-number: The journal entry with the assigned sequence number is the first entry considered for retrieval.

FROMTIME Parameter

Specifies the date and time of the first journal entry considered for retrieval.

date-time: The first journal entry found with the assigned time or the next later time found is the starting point for the search. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. If colons are used to separate the time values, then the value must be enclosed in apostrophes ('hh:mm:ss'). The format of the date must be as defined by the system values QDATFMT and, if separators are used, QDATSEP.

TOENT Parameter

Specifies the last journal entry considered for retrieval.

***LAST:** The search continues until the last journal entry in the journal receiver range specified is processed. If SEARCH(*DESCEND) is specified, TOENT(*LAST) is only valid if FROMENT(*LAST) is also specified.

***FIRST:** The search continues until the first journal entry in the journal receiver range specified is processed. If SEARCH(*ASCEND) is specified, TOENT(*FIRST) is only valid if FROMENT(*FIRST) is also specified.

ending-sequence-number: The journal entry with the assigned sequence number is the final entry considered for retrieval.

TOTIME Parameter

Specifies the date and time of the last entry considered for retrieval.

date time: The first journal entry found with the assigned time or the next earlier time found is the ending point for the search. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. If colons are used to separate the time values, then the value must be enclosed in apostrophes ('hh:mm:ss'). The format of the date must be as defined by the system values QDATFMT and, if separators are used, QDATSEP.

Note: The values specified for the FROM and TO parameter can be the same (for example, FROMENT(234) and TOENT(234) can be specified).

SEARCH Parameter

Specifies the order that the journal entries are searched to get an entry.

***ASCEND:** The journal entries are searched in ascending order (from the oldest entry to the newest entry).

***DESCEND:** The journal entries are searched in descending order (from the newest entry to the oldest entry).

JRNCDE Parameter

Specifies that the entries considered for retrieval are limited to the journal entries that contain the assigned journal code.

***ALL:** The search for the entry to get is not limited to a particular journal code.

***CTL:** The entries considered for retrieval are the journal entries created to control the Journal functions (journal codes J and F).

code: Specify the journal code that limits the search for the entry to retrieve. Only journal entries that contain the journal code are considered for retrieval. Up to six valid journal codes may be specified. Refer to the *Backup and Recovery Guide* for a list of valid journal codes.

ENTTYP Parameter

Specifies that the entries considered for retrieval are limited to the journal entries that contain the assigned entry type.

***ALL:** The search for the entry to get is not limited to a particular entry type.

***RCD:** The entries considered for retrieval are limited to journal entries created for record level operations.

entry-type: Specify the entry type that limits the search for the entry to get. Only journal entries that contain the specified entry type are considered for retrieval.

JOB Parameter

Specifies that the journal entries searched for retrieval are limited to the journal entries for a particular job. Only journal entries for the specified job are considered for retrieval.

***ALL:** The search is not limited to entries for a particular job.

***:** The search is limited to entries for the current job. Only entries for the requesting job are considered for retrieval.

job-name: Specify the name of the job whose journaled changes are considered for retrieval. Only journaled changes for this job are considered for retrieval. If no job name is given, all of the journal entries containing the simple name of the job are considered for retrieval.

PGM Parameter

Specifies that the journal entries considered for retrieval are limited to the journal entries created by a particular program. Only journal entries for the assigned program are considered for retrieval.

***ALL:** The search is not limited to entries for a particular program.

program-name: Specify the name of the program whose journaled changes are considered for retrieval. Only journaled changes for this program are considered for retrieval.

CMTCYCID Parameter

Specifies that the journal entries considered for retrieval are limited to the journal entries containing the specified commit cycle identifier.

***ALL:** The search is not limited to entries for a particular commit cycle identifier.

commit-cycle-identifier: Specify the commit cycle identifier that limits the search for the entry to get. Only journal entries that contain the commit cycle identifier are considered for retrieval.

Note: If more than one search value is specified, a journal entry must satisfy all of the search values in order to be retrieved. Search values are FILE, JRNCDE, ENTTYP, JOB, PGM and CMTCYCID. If none are specified, the first entry found that satisfies the values specified for RCVRNG, FROMENT, and TOENT is retrieved.

RTNSEQNBR Parameter

Specifies the name of the CL decimal variable in the program into which the journal entry sequence number of the retrieved journal entry is copied. If a CL variable name is not specified, the journal entry sequence number is not copied into the program. The specified variable must be a decimal variable that has a length of ten positions with no decimal positions.

RTNJRNCDE Parameter

Specifies the name of the CL character variable in the program into which the journal code of the retrieved journal entry is copied. If a CL variable name is not specified, the journal code of the retrieved journal entry is not copied into the program. The specified variable must be a character variable with a minimum length of 1 character. If the length of the variable is longer than 1, it is padded to the right with blanks.

RTNENTTYP Parameter

Specifies the name of the CL character variable in the program into which the entry type of the retrieved journal entry is copied. If a CL variable name is not specified, the entry type of the retrieved journal entry is not copied into the program. The specified variable must be a character variable with a minimum length of 2 characters. If the length of the variable is longer than 2, it is padded to the right with blanks.

RTNRCV Parameter

Specifies the name of the CL character variable in the program into which the journal receiver name from where the returned journal entry was retrieved is copied. If the CL variable name is not specified, the journal receiver name is not copied into the program. The specified variable must be a character variable with a minimum length of 10 characters. If the length of the variable is longer than 10, it is padded to the right with blanks.

RTNRCVLIB Parameter

Specifies the name of the CL character variable in the program that the journal receiver library name identified by the returned journal entry was put into is copied. If the CL variable name is not specified, the journal receiver library name is not copied into the program. The specified variable must be a character variable with a minimum length of 10 characters. If the length of the variable is longer than 10, it is padded to the right with blanks.

RTNJRNE Parameter

Specifies the name of the CL character variable in the program into which the retrieved journal entry is copied. The format of the retrieved journal entry is the same as a journal entry converted to an outfile by the Display Journal (DSPJRN) command. If a CL variable name is not specified, the retrieved journal entry is not copied into the program. The specified variable must be a character variable. If the retrieved journal entry is longer than the variable's field length, the entry is truncated. If the entry is shorter, it is padded to the right with blanks. Although this is a variable length field, the length of most journal entries is less than or equal to 125 characters, plus the length of the largest record image. The format of the fields in the retrieved entry is as follows:

```
ENTRY LENGTH          TYPE(*DEC) LEN(5 0)
SEQUENCE NUMBER       TYPE(*DEC) LEN(10 0)
JOURNAL CODE          TYPE(*CHAR) LEN(1)
JOURNAL ENTRY TYPE    TYPE(*CHAR) LEN(2)
DATE                  TYPE(*CHAR) LEN(6)
TIME                  TYPE(*DEC) LEN(6 0)
JOB NAME              TYPE(*CHAR) LEN(10)
USER NAME             TYPE(*CHAR) LEN(10)
JOB NUMBER            TYPE(*DEC) LEN(6 0)
PROGRAM NAME          TYPE(*CHAR) LEN(10)
OBJECT NAME           TYPE(*CHAR) LEN(10)
OBJECT LIBRARY        TYPE(*CHAR) LEN(10)
MEMBER NAME           TYPE(*CHAR) LEN(10)
COUNT/RRN             TYPE(*DEC) LEN(10 0)
FLAG                  TYPE(*CHAR) LEN(1)
COMMIT CYCLE ID       TYPE(*DEC) LEN(10 0)
RESERVED              TYPE(*CHAR) LEN(8)
ENTRY SPECIFIC DATA   TYPE(*CHAR) LEN(up to 1875)
```

Examples

Assume the following variables are specified:

```
DCL &SEQ        TYPE(*DEC) LENGTH(10,0)
DCL &JRNENT     TYPE(*CHAR) LENGTH(200)
DCL &RCVNAME    TYPE(*CHAR) LENGTH(10)
DCL &RCVLIB     TYPE(*CHAR) LENGTH(10)
```

and this command is run:

```
RTVJRNE   JRN(MYLIB/JRNA)  ENTTYP(PR)  RTNSEQNBR(&SEQ#)  +
  RTNJRNE(&JRNENT)
```

Since no starting journal entry is specified in this command, the first entry from the currently-attached receiver MYLIB/JRNA is considered. (The first entry in any receiver is always an identifier for the previously-attached receiver. This first receiver entry is known as a type PR entry, and it contains the name of the previously attached receiver in its entry-specific data.) The PR entry is the first entry in ascending order in the currently attached receiver; when it is found, the name of the previously attached receiver is from the entry-specific data portion of the retrieved PR entry and is placed into a CL variable named &JRNENT.

Then Change Variable (CHGVAR) commands can be used to separate the journal receiver name found in the entry specific data from the journal receiver library name as follows:

```
CHGVAR &RCVNAME  (%SST(&JRNENT 126 10))
CHGVAR &RCVLIB  (%SST(&JRNENT 136 10))
```

Assume the following variables are specified:

```
DCL &ENTNO        TYPE(*DEC) LEN(10 0)
DCL &JCODE        TYPE(*CHAR) LEN(1)
DCL &ETYPE        TYPE(*CHAR) LEN(2)
DCL &RCVNAME      TYPE(*CHAR) LEN(10)
DCL &RCVLIB       TYPE(*CHAR) LEN(10)
DCL &JENTRY       TYPE(*CHAR) LEN(205)
```

and this command is run:

```
RTVJRNE   JRN(MYLIB/JRNLA)  FILE(LIB1/A MBR3)  +
  RCVRNG(RCVLIB/RCV30 RCVLIB/RCV27)  +
  ORDER(*DESCEND)  JRNCDE(R)  ENTTYP(UP DL)  +
  JOB(000666/QPGMR/WORKST01)  PGM(TSTPGMA)  +
  RTNSEQNBR(&ENTNO)  RTNJRNCDE(&JCODE)  +
  RTNENTTYP(&ETYPE)  RTNRCV(&RCVNAME)  +
  RTNRCVLIB(&RCVLIB)  RTNJRNE(&JENTRY)
```

This command gets a journal entry, searching in descending order the journal receiver chain from receiver RCV30 in library RCVLIB to receiver RCV27 in library RCVLIB, journaled through journal JRNLA in library MYLIB, and copies the entry into the specified CL variables. The retrieved entry is an UPDATE or DELETE entry with journal code R from member MBR3 in file A in library LIB1, created in job 000666 QPGMR WORKST01 in program TSTPGMA. The sequence number of the retrieved entry is copied into CL variable &ENTNO. The journal code of the retrieved entry is copied into CL variable &JCODE. The entry type of the retrieved entry is copied into CL variable &ETYPE. The name of the journal receiver from which the returned entry was retrieved is copied into CL variable &RCVNAME. The library name of the journal receiver from which the returned entry was retrieved is copied into CL variable &RCVLIB. The whole journal entry that is retrieved is copied into CL variable &JENTRY.

RTVMSG (Retrieve Message) Command

The Retrieve Message (RTVMSG) command is used by a program to retrieve a specified predefined message from a message file and copy it into CL variables in the program. Substitution values can be specified in the MSGDTA parameter (as a single character string containing one or more concatenated message data fields) to replace the substitution variables in the predefined message text. The program can later write the message to an output device file to be printed, for example.

The CL prompt for this command lists the minimum length for retrieved variables next to the appropriate parameters. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length and the second number indicates the minimum number of decimal positions.

Restrictions: (1) This command is valid only in compiled CL programs. (2) To use this command, you must have *USE authority for the message file and the library in which the message is stored.

```
RTVMSG──MSGID message-identifier──────────────────────────────►

                                          ┌─┐
                                          │P│
►──MSGF──┬►*LIBL/────────┬──message-file-name┴─┴──────────────►
         ├─*CURLIB/──────┤
         └─library-name/─┘
                                                     Required
──────────────────────────────────────────────────────────────
                                                     Optional
                        ┌─┐
                        │P│
►──MSGDTA character-string┴─┴──MSG CL-variable-name───────────►

►──MSGLEN CL-variable-name──SECLVL CL-variable-name───────────►

►──SECLVLLEN CL-variable-name──SEV CL-variable-name───────────►

►──ALROPT CL-variable-name──LOGPRB CL-variable-name───────────
                                                 ┌──────────┐
                                                 │Pgm: B,I  │
                                                 └──────────┘
```

MSGID Parameter
Specifies the message identifier of the predefined message that is being retrieved from the specified message file.

MSGF Parameter
Specifies the qualified name of the message file that contains the predefined message being retrieved.

The possible library values are:

***LIBL:** The library list is used to locate the message file.

***CURLIB:** The current library for the job is used to locate the message file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message file is located.

message-file-name: Specifies the message identifer of the predefined message that is being retreived for the message file.

MSGDTA Parameter
Specifies, if the predefined message contains substitution variables, the substitution values that are used as the message data fields in the retrieved message. Either a character string or a CL variable containing the character string can be specified. As many substitution values can be specified in the character string as there are substitution variables in the predefined message. The values, which take the place of the substitution variables defined in the message text when the message was defined, must be specified by the following rules:

- Multiple values must be concatenated together to form a single character string. The length of the entire character string of concatenated substitution values cannot exceed 132 characters. The *CL Programmer's Guide* has more details.
- Multiple values must be specified in the same order in the character string as the substitution variables were defined in the MSG and FMT parameters of the Add Message Description (ADDMSGD) command.

- Each value must be specified as long as its associated variable was defined, and the specified character string must be the same length as the sum of the message data fields defined in the message.
- For multiple values, each substitution value can be specified as a CL program constant or CL variable; that is, any combination of constants and variables can be specified if they are first concatenated into one character string and they are in the same format and sequence expected.
- The length of the substitution value should be the same length as the length defined for the substitution variables. If the substitution value length is longer than the substitution variable length, the message data is truncated. If the substitution value is shorter, it becomes a null field.

Enter the character string that is to be used as the substitution value in the specified predefined message that is to be sent by the program, or enter the name of the CL variable that contains the character string. For more information on coding this parameter, refer to the description of the MSGDTA parameter in the SNDPGMMSG command.

MSG Parameter

Specifies the name of the CL character variable in the program into which the first-level text of the retrieved message is copied. If a CL variable name is not specified, the first-level text is not copied into the program.

The specified variable must be a character variable. If the retrieved first-level text is longer than the variable's field length, the text is truncated. If the text is shorter, it is padded to the right with blanks. Although this is a variable length field, most messages are designed to be less than 132 characters.

MSGLEN Parameter

Specifies the name of the CL decimal variable in the program into which the total length of the first-level text available to be retrieved is copied. The length specified is the total length after the substitution values (specified in the MSGDTA parameter) are placed in the first-level text.

The specified variable must be a decimal variable that has a length of five digits. If a CL variable name is not specified, the length is not copied into the program.

SECLVL Parameter

Specifies the name of the CL character variable in the program into which the second-level message of the retrieved message is copied. If a variable name is not specified, the second-level message is not copied into the program.

If the retrieved second-level message is longer than the variable's field length, the text is truncated. If the text is shorter, it is padded to the right with blanks. Although this is a variable length field, most second-level message is designed to be less than 3000 characters.

SECLVLLEN Parameter

Specifies the name of the CL decimal variable in the program into which the total length of the second-level message being retrieved is copied. The length specified is the total length after the substitution values (specified in the MSGDTA parameter) are placed in the second-level message.

The specified variable must be a decimal variable that has a length of five positions. If a CL variable name is not specified, the length is not copied into the program.

SEV Parameter

Specifies the name of the CL decimal variable into which the severity code of the retrieved message is copied. The specified variable must be a decimal variable that has a length of two positions. If a variable name is not specified, the severity code of the retrieved message is not copied into the program.

ALROPT Parameter

Specifies the name of the CL variable into which the alert option of the retrieved message is copied. The variable must be a character variable nine positions long.

| **LOGPRB Parameter**

| Specifies whether the message will be logged in the
| service activity log.

Examples

```
RTVMSG  MSGID(UIN0145)  MSGF(INVN)  MSG(&WORK)  +
  MSGDTA('any old time')
```

This command retrieves the message text of the message UIN0145 stored in the INVN message file. The retrieved text is copied into the CL variable &WORK after the substitution variables are replaced with the values *any*, *old*, and *time*. This example assumes that the substitution variables &1, &2, and &3 have been defined in the message as character variables, each 4 characters long.

```
RTVMSG  MSGID(UIN0150)  MSGF(INV)  MSG(&MSG)  +
  SECLVL(&SECLVL)
```

This command retrieves the first- and second-level message of the message UIN0150, which is stored in message file INV, and moves it into the CL variables &MSG and &SECLVL.

RTVNETA (Retrieve Network Attributes) Command

The Retrieve Network Attributes (RTVNETA) command is used in a CL program to retrieve the network attributes of the system.

The value(s) are returned (copied) to the specified CL variable(s) in the program. The following attributes can be retrieved:

- The current system name and pending system name.
- The local network ID.
- The local control point name.
- The default local location name.
- The default mode name.
- The maximum number of conversations for a remote location.
- The advanced peer-to-peer networking (APPN) nodetype.
- The maximum number of intermediate sessions.
- The route addition resistance.
- The list of APPN network node servers.
- The status of alerts.
- The status of the system as an alert primary focal point.
- The status of the system as an alert default focal point.
- The type of alert logging.
- The name of the alert controller description.
- The name of the default network message queue and the library that contains it.
- The name of the default network output queue and the library that contains it.
- The action to be taken for any job streams received through the network by the system.
- The maximum number of times in a SNADS network that a distribution queue entry may receive and reroute on the path to its final destination.
- The action taken for any distributed data management (DDM) file access requests received from other systems.
- The action to be taken for any PC Support (PCS) access requests.

Restrictions: (1) This command is valid only in a compiled CL program. (2) The attributes of the network attribute and the receiving CL variable must be compatible.

```
                                                                          Optional
RTVNETA──SYSNAME CL-variable-name──PNDSYSNAME CL-variable-name──────────────────►

►──LCLNETID CL-variable-name──LCLCPNAME CL-variable-name────────────────────────►

►──LCLLOCNAME CL-variable-name──DFTMODE CL-variable-name────────────────────────►

►──MAXLOCCNV CL-variable-name──NODETYPE CL-variable-name────────────────────────►

►──MAXINTSSN CL-variable-name───────────────────────────────────────────────────►

►──RAR CL-variable-name──NETSERVER CL-variable-name─────────────────────────────►

►──ALRSTS CL-variable-name──ALRPRIFP CL-variable-name───────────────────────────►

►──ALRDFTFP CL-variable-name──ALRLOGSTS CL-variable-name────────────────────────►

►──ALRCTLD CL-variable-name──MSGQ CL-variable-name──────────────────────────────►

►──MSGQLIB CL-variable-name──OUTQ CL-variable-name ─────────────────────────────►

►──OUTQLIB CL-variable-name──JOBACN CL-variable-name────────────────────────────►

►──MAXHOP CL-variable-name──────────────────────────────────────────────────────►

►──DDMACC CL-variable-name──DDMACCLIB CL-variable-name──────────────────────────►

►──PCSACC CL-variable-name──PCSACCLIB CL-variable-name───────────────────────────
                                                              ┌────────────────┐
                                                              │Pgm: B,I        │
                                                              └────────────────┘
```

SYSNAME Parameter
Specifies the name of the CL variable that receives the current system name. You must specify the name of a character variable with a minimum length of 8 characters. (If the system name has fewer characters than the variable allows, the value is padded on the right with blanks.)

PNDSYSNAME Parameter
Specifies the name of the CL variable that receives the pending system name. You must specify the name of the character variable with a minimum length of 8 characters. The value returned is blank if there is no pending system name. (If the system name has fewer characters than the variable allows, the value is padded on the right with blanks.)

LCLNETID Parameter
Specifies the name of the CL variable that receives the current network ID. You must specify the name of the character variable with a minimum length of 8 characters. (If the network ID has fewer characters than the variable allows, the value is padded on the right with blanks.)

LCLCPNAME Parameter
Specifies the name of the CL variable that receives the current local control point. You must specify the name of the character variable with a minimum length of 8 characters. (If the control point has fewer characters than the variable allows, the value is padded on the right with blanks).

LCLLOCNAME Parameter
Specifies the name of the CL variable that receives the default local location name. You must specify the name of the character variable with a minimum length of 8 characters. (If the local location has fewer characters than the variable allows, the value is padded on the right with blanks.)

DFTMODE Parameter
Specifies the name of the CL variable that receives the default mode name. You must specify the name of the character variable with a minimum length of 8 characters. (If the default mode has fewer characters than the variable allows, the value is padded on the right with blanks.)

MAXLOCCNV Parameter
Specifies the name of the CL variable that receives the maximum number of conversations for a remote location. You must specify the name of the decimal variable with a minimum length of 5 digits and no decimal points.

NODETYPE Parameter
Specifies the name of the CL variable that receives the current APPN nodetype. You must specify the name of the character variable with a minimum length of 8 characters. The following values may be returned in a CL variable:

***ENDNODE:** Specifies that the node does not provide network services to other nodes but may participate in the APPN network by using the services of an attached network server or may operate in a peer environment similar to migration end nodes.

***NETNODE:** Specifies that the node provides intermediate routing, route selection services, and distributed directory services for local users and to end nodes and migration end nodes that it is serving.

MAXINTSSN Parameter
Specifies the name of the CL variable that receives the maximum number of intermediate sessions. You must specify the name of the decimal variable with a minimum length of 5 digits and no decimal positions.

RAR Parameter
Specifies the name of the CL variable that receives the route addition resistance. You must specify the name of the decimal variable with a minimum length of 5 digits and no decimal positions.

NETSERVER Parameter
Specifies the name of the CL variable that receives the list of network node servers. You must specify the name of a character variable with a minimum length of 85 characters. (If the server control point name or network ID has fewer characters than the variable allows, the value is padded on the right with blanks). The list can contain five node servers. Each server has the form: Network ID (9 characters) followed by the server name (8 characters). There are no separators. The network ID may contain the value *LCLNETID which specifies that the current network ID is used. If there are less than five node servers specified, the remaining ones contain hexadecimal zeros for a name. As soon as the first null name is encountered in the list, it is safe to assume that all the remaining names will also be null.

ALRSTS Parameter
Specifies the name of the CL variable that receives the current system alert status. You must specify the name of a character variable with a minimum length of 10 characters. (If the alert status value has fewer characters than the variable allows, the value is padded on the right with blanks.) The following values can be returned in the CL variable:

***ON:** Alert processing is currently active.

***UNATTEND:** Alert processing is currently active but the system is unattended.

***OFF:** Alert processing is not currently active.

ALRPRIFP Parameter
Specifies if the system is a primary alert focal point. You must specify the name of a character variable with a minimum length of 10 characters. (If the primary focal point value has fewer characters than the variable allows, the value is padded on the right with blanks.) The following values can be returned in the CL variable:

***NO:** The system is not an alert primary focal point.

***YES:** The system is an alert primary focal point.

ALRDFTFP Parameter
Specifies if the system is a default alert focal point. You must specify the name of a character variable with a minimum length of 10 characters. (If the primary focal point value has fewer characters than the variable allows, the value is padded on the right with blanks.) The following values can be returned in the CL variable:

***NO:** The system is not an alert default focal point.

***YES:** The system is an alert default focal point.

ALRLOGSTS Parameter
Specifies the name of the CL variable that receives the current alert logging status. You must specify the name of a character variable with a minimum length of 10 characters. The following values can be returned in the CL variable:

***NONE:** Do not log alerts.

***LOCAL:** Log only locally generated alerts.

***RCV:** Log only alerts received from other nodes.

***ALL:** Log both locally generated and incoming alerts.

ALRCTLD Parameter
Specifies the name of the CL variable that receives the name of the controller through which alert messages are sent to another system when alert processing is active. You must specify the name of a character variable with a minimum length of 10 characters. (If the alert controller name has fewer characters than the variable allows, the value is padded on the right with blanks.) The following value can be returned in the CL variable:

***NONE:** There is no controller for alerts.

MSGQ Parameter
Specifies the name of the CL variable that receives the system default network message queue name. You must specify the name of a character variable with a minimum length of 10 characters. (If the message queue name has fewer characters than the variable allows, the value is padded on the right with blanks.)

MSGQLIB Parameter
Specifies the name of the CL variable that receives the name of the library that contains the system-default network message queue. You must specify the name of a character variable with a minimum length of 10 characters. (If the library name has fewer characters than the variable allows, the value is padded on the right with blanks.)

OUTQ Parameter
Specifies the name of the CL variable that receives the system default network output queue name. You must specify the name of a character variable with a minimum length of 10 characters. (If the output queue name has fewer characters than the

variable allows, the value is padded on the right with blanks.)

OUTQLIB Parameter
Specifies the name of the CL variable that receives the name of the library that contains the system-default network message queue. You must specify the name of a character variable with a minimum length of 10 characters. (If the library name has fewer characters than the variable allows, the value is padded on the right with blanks.)

JOBACN Parameter
Specifies the name of the CL variable that receives the current job action for job streams received through the network. You must specify the name of a character variable with a minimum length of 10 characters. (If the job action value has fewer characters than the variable allows, the value is padded on the right with blanks.) The following values can be returned in the CL variable:

***FILE:** Job streams received from the network are filed in the queue of network files for the user it was sent to.

***REJECT:** Job streams received from the network are rejected.

***SEARCH:** The table of network job entries is searched to determine the action for any job streams received.

MAXHOP Parameter
Specifies the name of the CL variable that receives the maximum number of systems in the SNADS network that a distribution queue originating at this node may be received and rerouted on the path to its final destination. You must specify the name of a decimal variable with a total length of 5 digits, and no decimal positions.

DDMACC Parameter
Specifies the name of the CL variable that receives the current system action for DDM requests from other systems. You must specify the name of a character variable with a minimum length of 10 characters. (If the DDM access value has fewer characters than the variable allows, the value is padded on the right with blanks.) The following values can be returned in the CL variable:

***REJECT:** The system does not allow any DDM requests from remote systems. However, the system may use DDM to access files on remote systems. (A remote system cannot access any file on a system that specifies *REJECT).

***OBJAUT:** All file requests are allowed and controlled by the object authorizations on the system (normal system object level security).

program-name: The name of the customer validation program that can supplement system object level security. This user-exit program can restrict user access to *PUBLIC and private files. The target DDM support calls the user program for each reference to a file. The user-exit program indicates to DDM if the request should proceed or end.

DDMACCLIB Parameter
Specifies the name of the CL variable that receives the name of the library that contains the DDM access program. You must specify the name of a character variable with a minimum length of 10 characters. (If the library name has fewer characters than the variable allows, the value is padded on the right with blanks.) If *REJECT or *OBJAUT is returned for the DDMACC parameter, the value for DDMACCLIB will be all blanks.

PCSACC Parameter
Specifies the name of the CL variable that receives the current system action for PCS requests. You must specify the name of a character variable with a minimum length of 10 characters. (If the PCS access value has fewer characters than the variable allows, the value is padded on the right with blanks.) The following values can be returned in the CL variable:

***REJECT:** The system does not allow any PCS requests.

***OBJAUT:** All PCS requests are allowed and controlled by the object authorizations on the system.

program-name: The name of the customer supplied PCS host system application exit program that can supplement system object level security.

PCSACCLIB Parameter
Specifies the name of the CL variable that receives the name of the library that contains the PCS access program. You must specify the name of a character variable with a minimum length of 10 characters. (If the library name has fewer characters than the variable allows, the value is padded on the right with blanks.) If *REJECT or *OBJAUT is returned for the PCSACC parameter, the value for PCSACCLIB is all blanks.

Example

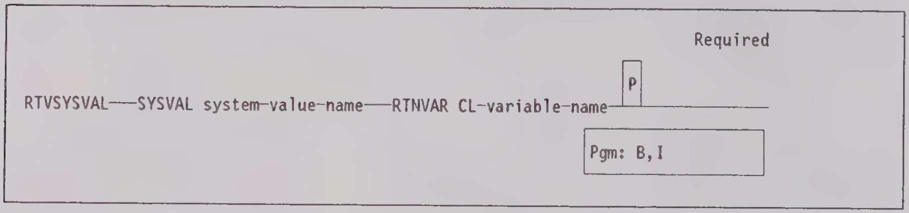
```
DCL  VAR(&SNAME)  TYPE(*CHAR)  LEN(8)  +
  DCL  VAR(&HOPS)  TYPE(*DEC)  LEN(5 0)  +
  RTVNETA  SYSNAME(&SNAME)  MAXHOP(&HOPS)
```

This command retrieves the current system name and the maximum number of systems in a SNADS network that a distribution queue entry may be received and rerouted on the path to its destination.

RTVSYSVAL (Retrieve System Value) Command

The Retrieve System Value (RTVSYSVAL) command is used in a CL program to retrieve the value from the specified system value so that it can be used in the program. The value is returned (copied) to the specified CL variable in the program.

Restrictions: (1) This command is valid only in compiled CL programs. (2) The attributes of the system value and the receiving CL variable must be compatible.



SYSVAL Parameter

Specifies the name of the system value whose value is retrieved and returned for use in the program. The *Work Management Guide* has the names and descriptions of the system values that can be specified.

Double-byte character set considerations:

In addition to the system values specified in the *Work Management Guide*, you can retrieve the value of QIGC, which indicates if you installed the double-byte character version of the AS/400 system. If the system value for QIGC is 1, then the double-byte character version of the AS/400 system is installed on your system. If the system value for QIGC is 0, the double-byte character version of the AS/400 system is not installed on your system.

RTNVAR Parameter

Specifies the name of the CL program variable that receives the value of the system value being returned. The type and length for the CL variable when it was declared must be compatible with that of the system value whose value is received. (The *Work Management Guide* describes the attributes of individual system values.)

In general, the type of the return variable must match the type of the system value. For character system values that are 1 character long, the CL variable can be a character or logical variable. For character and logical system values, the length of the CL variable must equal the length of the system value. For decimal system values, the CL variable must have a length that is greater than or equal to the length of the system value.

Example

```
RTVSYSVAL  SYSVAL(QDATE)  RTNVAR(&DATE)
```

This command retrieves the date value from the system value QDATE and copies it into the CL variable &DATE. The CL variable must be declared as a 6-character variable to match the attributes of the QDATE system value.

RTVUSRPRF (Retrieve User Profile) Command

The Retrieve User Profile (RTVUSRPRF) command is used in a control language (CL) program to get one or more of the values that are stored and associated with a user. The values are returned in the specified CL variables for the desired user.

The CL prompt for this command lists the minimum length for the variables next to the appropriate parameters you want to get. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length and the second number indicates the minimum number of decimal positions.

Restriction: *READ authority is required for the user specified on the USRPRF parameter if any of the following parameters are specified: SPCAUT, MAXSTG, PTYLMT, GRPPRF, USRCLS, OWNER, GRPAUT, ACGCDE, TEXT, or PWDCHGDAT. If one of the above parameters is specified, and the job does not have *READ authority to the specified user, none of the values requested are returned.

```
                                                                 Optional
                           ┌►*CURRENT──┐  ┌─┐
                           │           │  │P│
RTVUSRPRF──USRPRF──────────┤           ├──┴─┴──RTNUSRPRF CL-variable-name──────────►
                           └─user-name─┘

►──SPCAUT CL-variable-name─────────────────────────────────────────────────────────►

►──MAXSTG CL-variable-name──PTYLMT CL-variable-name────────────────────────────────►

►──INLPGM CL-variable-name──INLPGMLIB CL-variable-name─────────────────────────────►

►──JOBD CL-variable name──JOBDLIB CL-variable-name─────────────────────────────────►

►──GRPPRF CL-variable-name──OWNER CL-variable-name─────────────────────────────────►

►──GRPAUT CL-variable-name──ACGCDE CL-variable-name──MSGQ CL-variable-name─────────►

►──MSGQLIB CL-variable-name──OUTQ CL-variable-name─────────────────────────────────►

►──OUTQLIB CL-variable-name──TEXT CL-variable-name─────────────────────────────────►

►──PWDCHGDAT CL-variable-name──USRCLS CL-variable-name─────────────────────────────►

►──SPCENV CL-variable-name──CURLIB CL-variable-name────────────────────────────────►

►──INLMNU CL-variable-name──INLMNULIB CL-variable-name─────────────────────────────►

►──LMTCPB CL-variable-name──DLVRY CL-variable-name──SEV CL-variable-name───────────►

►──PRTDEV CL-variable-name──ATNPGM CL-variable-name────────────────────────────────►

►──ATNPGMLIB CL variable-name──USROPT CL-variable-name─────────────────────────────
                                                              ┌──────────────┐
                                                              │Pgm: B,I      │
                                                              └──────────────┘
```

USRPRF Parameter
Specifies the name of the user whose information you want to get. If a variable is specified, it must be 10 characters in length and contain a user name or the value *CURRENT.

__*CURRENT:__ The information returned in the specified parameters are for the user of the current job.

user-name: Specify the name of the user for information that you want to get.

RTNUSRPRF Parameter
Specifies the name of a 10-character CL variable that is used to get the name of the user. If *CURRENT is specified on the USRPRF parameter, the value returned is the user name of the current job; if a user name is specified, that name is returned for this parameter.

SPCAUT Parameter

Specifies the name of a 100-character CL variable that is used to get the list of special authorities the user has. The format returned is a list of up to 10 entries, with each entry 10 characters in length; each entry contains a special authority. If there are fewer than 10 special authorities in the list, the remaining entries are padded on the right with blanks. If the user has no special authorities, the first entry contains the value of *NONE followed by blanks.

The special authority list is returned in the following format:

```
Entry-1
  Special authority CHAR(10)
Entry-2
  Special authority CHAR(10)
   .
   .
   .
Entry-10
  Special authority CHAR(10)
```

MAXSTG Parameter

Specifies the name of a decimal CL variable length of (11 0) that is used to get the maximum amount of auxiliary storage that can be assigned to store permanent objects owned by the specified user. The value returned is either an 11-digit value or a value of -1 if it is *NOMAX. The variable must be an 11-digit value with zero decimal positions (11 0). (For additional information on this parameter, see the Create User Profile (CRTUSRPRF) command.)

PTYLMT Parameter

Specifies the name of a 2-character CL variable length that is used to get the highest scheduling priority that the user is allowed to have for each job that is submitted to the system. This value controls the job processing priority that any job running under this user can have; that is, values specified in the JOBPTY and OUTPTY parameters of any job command cannot exceed the PTYLMT value of the user under which the job is run. The scheduling priority can have a value of 0 through 9, where 0 is the highest priority and 9 is the lowest.

INLPGM Parameter

Specifies the name of a 10-character CL variable that is used to get the name of the initial program that starts when the specified user signs on the system. If no first program name is associated with the specified user, the value returned in the CL variable is *NONE. (For more information on this parameter, see the Create User Profile (CRTUSRPRF) command.)

INLPGMLIB Parameter

Specifies the name of a 10-character CL variable that is used to get the name of the library which contains the initial program associated with the specified user. If there is no initial program associated with the specified user, blanks are returned in the CL variable.

JOBD Parameter

Specifies the name of a 10-character CL variable that is used to get the name of the job description associated with the specified user.

JOBDLIB Parameter

Specifies the name of a 10-character CL variable that is used to get the name of the library that contains the job description that is associated with the specified user.

GRPPRF Parameter

Specifies the name of a 10-character CL variable that is used to get the name of the group profile. The group value specifies the name of the user whose authorities are made available to this user without granting authority for each object. If no group profile exists for the specified user profile, a value of *NONE is returned in the CL variable. (For more information on this parameter, see the CRTUSRPRF (Create User Profile) command.)

OWNER Parameter

Specifies the name of a 10-character CL variable that is used to get the special value of *USRPRF or *GRPPRF. This parameter indicates the user that is the owner of newly created objects: either the specified user or the group profile. If no group profile exists for the specified user profile, the value returned in the CL variable is *USRPRF.

GRPAUT Parameter

Specifies the name of a 10-character CL variable that is used to get the type of authority granted to the group profile for newly created objects. The special value of *CHANGE, *ALL, *USE, or *EXCLUDE is returned in the CL variable. If there is no group profile for the specified user, the special value of *NONE is returned; further, if the group profile is the owner of the objects the specified user creates, the special value returned is *NONE. (For more information on this parameter, see the CRTUSRPR (Create User Profile) command.)

ACGCDE Parameter

Specifies the name of a 15-character CL variable that is used to get the value of the accounting code assigned to the specified user. If no accounting code exists for the user profile, blanks are returned to the user. (For more information on this parameter, see the Create User Profile (CRTUSRPRF) command.)

MSGQ Parameter

Specifies the name of a 10-character CL variable that is used to get the name of the message queue associated with the specified user. (For more information on this parameter, see the CRTUSRPRF (Create User Profile) command.)

MSGQLIB Parameter

Specifies the name of a 10-character CL variable that is used to get the name of the library containing the message queue that is associated with the specified user.

OUTQ Parameter
Specifies the name of a 10-character CL variable that is used to get the name of the output queue associated with the specified user. The value of *DEV is returned for OUTQ if *DEV is the current value for this parameter.

OUTQLIB Parameter
Specifies the name of a 10-character CL variable that is used to get the name of the library containing the output queue that is associated with the specified user. Blanks are returned if the current value for OUTQ is *DEV.

TEXT Parameter
Specifies the name of a 50-character CL variable that is used to get the user-defined description for the specified user profile. If there is no text associated with the user, blanks are returned in the CL variable.

PWDCHGDAT Parameter
Specifies the name of a 10-character CL variable that is used to get the date of the last time the password for the specified user was changed. The variable specified must be 6 characters in length. The date is returned in the form YYMMDD. If the user does not have a date, blanks are returned.

USRCLS Parameter
Specifies the name of a 10-character CL variable that is used to get the user class for the specified user. The special value of *USER, *SYSOPR, *PGMR, *SECADM, or *SECOFR is returned as the CL variable.

SPCENV Parameter
Specifies the name of a 10-character CL variable that is the starting environment for the specified user. The special value of *SYSVAL, *NONE or *S36 is returned in the CL variable.

CURLIB Parameter
The name of the 10-character variable that is used to get the name of the job's default library for the specified user. A value of *CRTDFT is returned in the variable if no current library exists for this user.

INLMNU Parameter
Specifies the name of a 10-character CL variable that is used to get the name of the user's first initial when the specified user signs on the system. (For more information on this parameter, see the CRTUSRPRF (Create User Profile) command.)

INLMNULIB Parameter
The name of the 10-character variable that is used to get the library name containing the initial menu.

LMTCPB Parameter
Specifies the name of a 10-character CL variable that is used to get the value for the limited capability of a user to change his user profile. The value of *NO, *YES, or *PARTIAL is returned in the CL variable.

DLVRY Parameter
Specifies the name of a 10-character CL variable that is used to get the message control delivery value for the specified user profile. The special value of *NOTIFY, *BREAK, *HOLD, or *DFT is returned in the CL variable.

SEV Parameter
Specifies the name of a decimal CL variable length of (2 0) that is used to get the message control severity level for the specified user.

PRTDEV Parameter
Specifies the name of a 10-character CL variable that is used to get the name of the printer device for the specified user. A value of *SYSVAL is returned if the printer device name is from the system value QPRTDEV.

ATNPGM Parameter
Specifies the name of a 10-character CL variable that is used to get the name of the attention-key-handling program for the specified user. A value of *NONE is returned if no attention-key-handling program is specified for this user.

ATNPGMLIB Parameter
Specifies the name of a 10-character CL variable that is used to get the name of the library of the attention key handling program for the specified user. If *NONE is the current value of ATNPGM, blanks are returned in the CL variable.

USROPT Parameter
Specifies the name of a 240-character CL variable that is used to get the list of user option values for the specified user. The special values of *NONE or a list of values is returned in the CL variable.

Example

The user enters this command:

```
CRTUSRPRF  USRPRF(SMITH)  SPCAUT(*SAVSYS *SECADM)  +
  MAXSTG(*NOMAX)  PTYLMT(4)  INLPGM(*NONE)  +
  MSGQ(QGPL/SMITHMQ)  OUTQ(QGPL/QSMITH)  +
  TEXT('John Smith User Profile')
```

When user Smith calls a CL program containing the following:

```
DCL     &UNAME       *CHAR 10
DCL     &URIGHT      *CHAR 100
DCL     &IPGM        *CHAR 10
DCL     &IPGMLB      *CHAR 10
DCL     &UMSGQ       *CHAR 10
DCL     &UMSQLB      *CHAR 10
```

The following values are returned:

```
RTVUSRPRF  USRPRF(*CURRENT)  RTNUSRPRF(&UNAME)  SPCAUT(&URIGHT) +
  INLPGM(&IPGM)  INLPGMLIB(&IPGMLB)
```

RUNQRY (Run Query) Command

The Run Query (RUNQRY) command is used to run an existing query or to run a default query if only a file name is specified for this command. The query gets information from the system data base and produces a report of that information. The report is created in either detailed or summary form. The definition of the query can be printed when ouput to a printer or data base file is specified. The output is displayed, printed, or stored in another data base file.

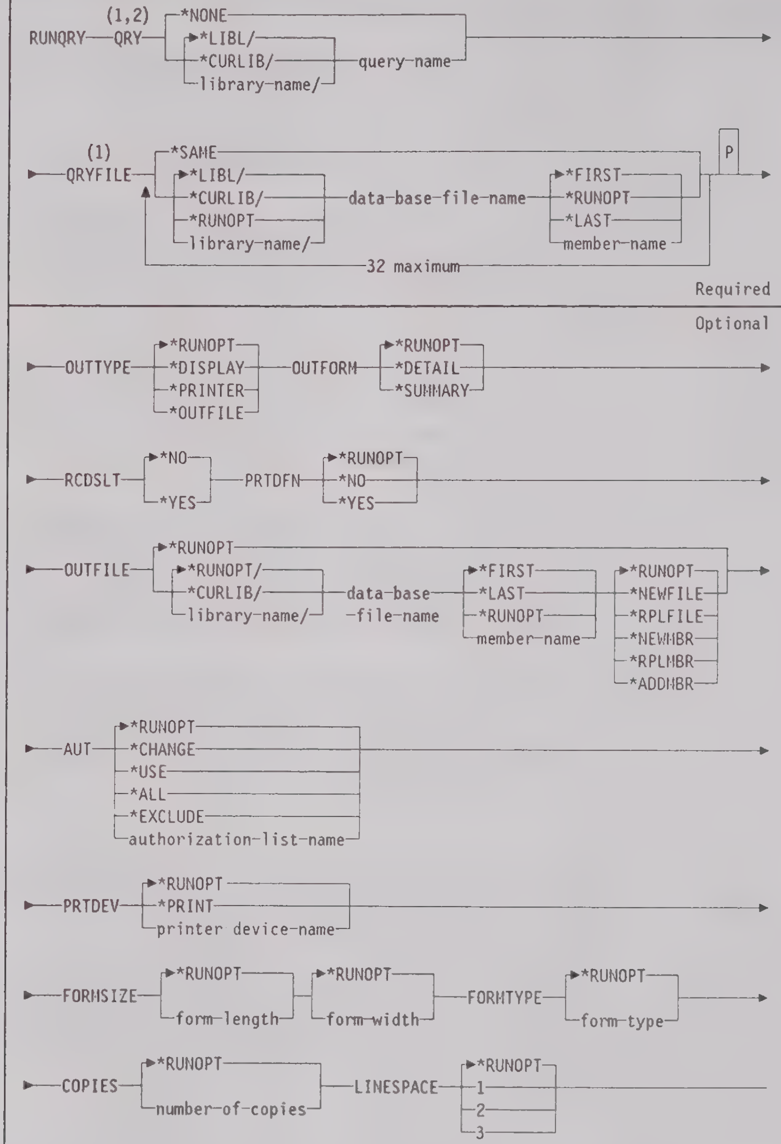
The command can be used in three ways: to run an existing query (one that has already been created), to run an existing query with some of its values changed by values specified on this command, or to run a default query based only on the values specified on this command.

- If you want to use an existing query without changing the file or files to query, use the QRY parameter (without the QRYFILE parameter) to specify the name of the query.
- If you want to query a file and have no previously defined query definition, use the QRYFILE parameter to specify which file to query; only *one* file name can be specified for a default query.
- If you want to run a changed version of an existing query, use the QRY parameter and the appropriate parameters to change the definition as desired. The parameter values specified on this command override the corresponding values in the existing query definition, but only when the command is processing. For example, use the QRYFILE parameter if you want to indicate a different file or list of files to use in the query.

If you specify both the QRY and QRYFILE parameters, the files specified in the QRYFILE parameter override the file name(s) specified in the query. Therefore, if multiple files (and members) are defined in the query definition and you want to change one or two of them, you must specify *SAME for the file selections that do not change and specify the values for the ones you want to override.

Notes:

1. When you run a changed version of an existing query, the changes you specify on the RUNQRY command do *not* change any of the values in the query definition itself; they affect only the results of the report being run.
2. If an existing query (identified on the QRY parameter) is used, *RUNOPT is the default value for any unspecified parameters; that is, the same value specified (or assumed) in the definition of the query is used as the default. If this is a default query, the default value is not *RUNOPT, but is the next predefined value listed in the syntax diagram, if any.



(1) Either QRY or QRYFILE must be specified; or, both can be specified.
(2) *RUNOPT is the default value for the remaining parameters. The parameter will default to whatever was specified in the query definition if a query name is entered. The default will be the second in the list if only a file name is entered.

Job: B,I Pgm: B,I Exec

QRY Parameter

Specifies the name of an existing query to run. If QRY is not specified, QRYFILE must be specified.

***NONE:** No existing query definition is used; instead, a default query (or quick query) is used to get information from the file specified on the QRYFILE parameter.

If the QRYFILE parameter is not specified, the file or files that were specified when the query was defined are the files used to run the query.

The possible library values are:

***LIBL:** The library list is used to locate the specified query definition.

***CURLIB:** The current library for the job is used to locate the specified query definition. If no current library entry is found in the library list, library QGPL is used.

library-name: Specify the library where the specified query is located.

query-name: Specify the qualified name of the query to run.

QRYFILE Parameter

Specifies the data base file or files where the system gets information. If the QRY parameter is specified, as many as 32 files can be specified on this parameter by using the file names and/or using the default value *SAME for one or more of the files. If the QRY parameter is not specified, only one file name can be specified on this parameter. If QRYFILE is not specified, QRY must be specified.

***SAME:** If a query definition is being used, the file or list of files specified in the query definition is the file or files used to produce the query report.

The possible library values are:

***LIBL:** The library search list is used to locate the data base file.

***CURLIB:** The current library is used to locate the specified data base file. If the specified current library is not found in the library list, then library QGPL is used.

***RUNOPT:** The library specified for the file selection is used.

library-name: Specify the name of the library to be searched.

data-base-file-name: Specify the names of one or more data base files that contain the data the system gets information from to produce the output. A maximum of 32 files can be specified.

The possible member values are:

***FIRST:** The first member in the file is the query member.

***RUNOPT:** The member specified for this file selection is used.

***LAST:** The last member in the file is the query member.

member-name: Specify the name of the file member to be the query member.

OUTTYPE Parameter

Specifies where the report or output produced by the query is sent. If no parameter is specified, the output device that was specified when the query was defined is assumed. If no output device was specified in the query, or if you do not specify a query name, *DISPLAY is assumed.

***RUNOPT:** If a query definition is being used, the type of output specified in the query definition is the type produced when this query is run.

***DISPLAY:** The output produced by the query is sent to the display station that runs the command.

***PRINTER:** The output produced by the query is printed.

***OUTFILE:** The output produced by the query is written to a data base file.

OUTFORM Parameter

Specifies the form of output to be produced by the query. If no parameter is specified, the *DETAIL or *SUMMARY value that was specified in the query is assumed. If no value was specified in the query and not entered on the command, or if a query name is not specified, *DETAIL is assumed.

***RUNOPT:** If a query definition is being used, the output form specified in the query definition is used when this query is run.

***DETAIL:** The output form produced by the query is a report containing detail records and summary records if any exist.

***SUMMARY:** The output form produced by the query is a report containing summary records only.

RCDSLT Parameter

Specifies if the query is or is not run with a run time selection test.

***NO:** The query is run without showing the record selection displays in AS/400 Query.

***YES:** Record selection definition is allowed for this run only. A display is shown on which you can change the record selection tests defined in the query or specify record selection tests if a query name was not specified.

PRTDFN Parameter

Specifies if the query definition is or is not printed with the report when the query is run. The definition can be printed when the output of the query is printed or stored in a data base file, as determined by the OUTTYPE parameter.

If no parameter is specified, the *NO or *YES value that was specified in the query is assumed. If no value was specified in the query, or if a query name is not specified, *NO is assumed.

*RUNOPT: If a query definition is being used, the print default option specified in the query definition is the type produced when this query is run.

***NO:** The query definition is not printed when the query is run.

***YES:** The query definition is printed after the report is printed. *YES cannot be specified if OUTTYPE(*DISPLAY) is specified or assumed.

OUTFILE Parameter

Specifies the data base file (if any) that receives the query output. If no parameter is specified, the library, file, and member that was specified in the query is assumed. If the name specified as the OUTFILE does not exist, the system creates it in the specified library.

*RUNOPT: If specified in the query definition, the output is directed to the data base file named in the query definition. The default data base file QQRYOUT located in library QGPL, member *FIRST, is used to store the query output if no OUTFILE value is specified.

The possible library values are:

*RUNOPT: The library specified in the query is used to locate the data base file.

***CURLIB:** The current library for the job is used to locate the data base file. If no current library exists in the library list, library QGPL is used.

library-name: Specify the library where the data base file is located.

data-base-file-name: Specify the qualified name of the data base file that receives the output of the command.

The possible member values are:

*FIRST: The first member in the file is used to receive the query output.

***LAST:** The last member in the file is used to receive the query output.

***RUNOPT:** The member specified in the query is used to receive the query output.

member-name: Specify the qualified name of the file member used to receive the query output.

The last option specifies whether to write the file to a new data base file, to replace an existing data base file, to add a new member, to replace an existing memter, or to add data to an existing member. If no parameter is specified, the value that was specified in the query is assumed. If no value is specified in the query, or if you do not specify a query name, NEWFILE is assumed.

The possible option values are:

*RUNOPT: If a query definition is being used, the member option specified in the query definition is the type used when this query is run.

***NEWFILE:** The output is written to a new data base file.

***RPLFILE:** The output deletes the old file and creates a new file.

***NEWMBR:** The output is added as a new member.

***RPLMBR:** The existing member is cleared and the output is then added.

***ADDMBR:** The output is added to the end of an existing member.

AUT Parameter

Specifies the authority for the outfile if one is specified in the query or specified in the OUTFILE parameter.

*RUNOPT: If specified in the query definition, the authority named in the query definition is used.

***CHANGE:** Change authority allows the user to perform all operations on the file except those limited to the owner or controlled by object existence authority and object management authority. Change authority provides object operational authority and all data authorities. When only QRYFILE is specified and the AUT parameter is specified or is *RUNOPT, *CHANGE is the default.

***ALL:** All authority allows the user to perform all operations on the file except those limited to the owner or controlled by authorization list management authority.

***USE:** Use authority allows the user to perform basic operations on the file. Use authority provides object operational authority and read authority.

***EXCLUDE:** Exclude authority prevents the user from accessing the file.

authorization-list-name: Specify the name of the authorization list whose authority is used for the file.

PRTDEV Parameter

Specifies the printer device on which the report is printed. If no parameter is specified, the printer that was specified when the query was defined is assumed. If no printer was specified in the query, or if you do not specify a query name, the session printer is assumed.

*RUNOPT: If a query definition is being used, the printer specified in the query definition is used to print the output when this query is run.

***PRINT:** The default printer is used to print the output when this query is run if the PRTDEV value is left blank.

printer-device-name: Specify the name of the printer that is used to print the output when this query is run.

FORMSIZE Parameter

Specifies the length and width of the forms on which the report is printed. If no parameter is specified, the form width that was specified in the query is assumed. If no form width was specified in the query, or if you do not specify a query name, 132 is assumed.

***RUNOPT:** If a query definition is being used, the form size specified in the query definition is used when this query is run.

form-length: Specify the form length used when this query is run. A value of 60 to 198 can be specified.

form-width: Specify the form width used when this query is run. A value of 60 to 198 can be specified.

FORMTYPE Parameter

Specifies the name of the particular form on which you want the report printed. When the report is ready to print, a message at the subconsole asks if that particular form is on the printer. Then the operator can check the printer and change forms, if necessary, before printing.

The forms ID cannot contain the following characters: commas (,), apostrophes ('), blanks, question marks (?), slashes (/), greater than signs (>), equal signs (=), and hyphens (-). If no parameter is specified, the forms ID that was specified in the query is assumed. If no number was specified in the query, or if you do not specify a query name, no number is assumed.

***RUNOPT:** If a query definition is being used, the form type specified in the query definition is used when this query is run.

form-type: Specify the form type that is used when this query is run. You can enter from 1 through 4 characters.

COPIES Parameter

Specifies the number of copies of the report to be printed. If no value for the parameter is specified, the number of copies that was specified in the query is assumed. If no number was specified in the query, or if you do not specify a query name, 1 is assumed.

***RUNOPT:** If a query definition is being used, the number of copies specified in the query definition is used when this query is run.

number-of-copies: Specify the number of copies to be printed when this query is run. You can specify a number from 1 through 255.

LINESPACE Parameter

Specifies the number of blank lines to leave between lines in the report. You can enter a number from 1 through 3. If no parameter is specified, the line spacing that was specified in the query is assumed. If no line spacing was specified in the query, or if you do not specify a query name, 1 is assumed.

***RUNOPT:** If a query definition is being used, the number of lines specified in the query definition is used when this query is run.

1: Indicates single spacing (no blank lines).

2: Indicates double spacing (1 blank line) is used when the query output is printed.

3: Indicates triple spacing (2 blank lines) is used when the query output is printed.

Examples

```
RUNQRY  QRY(LIBX/QRY1)  OUTYPE(*PRINTER)  OUTFORM(*SUMMARY)  COPIES(4)
```

This command runs the query QRY1 located in library LIBX. The report produced and printed contains summary records only. Four copies of the report are printed.

```
RUNQRY  QRYFILE(LIBX/FILE2 *FIRST)  OUTYPE(*OUTFILE)  OUTFORM(*DETAIL)  +
  RCDLST(*YES)  OUTFILE(LIB2/OUT1 MBR4)  MBROPT(*NEWMBR)
```

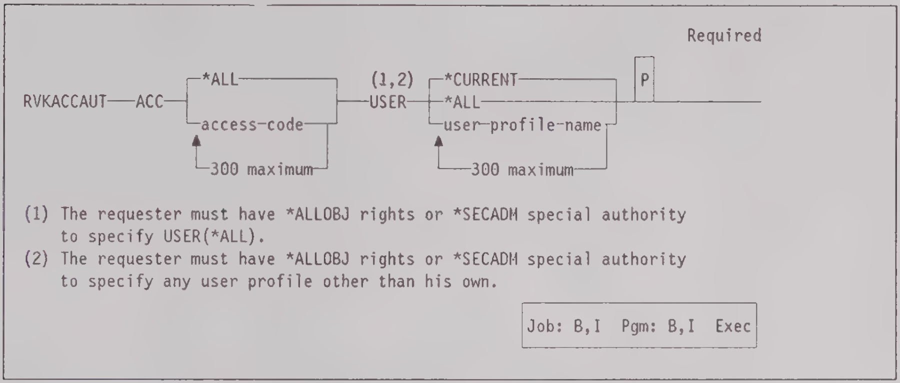
This command runs a default query and gets the data from the first member of file FILE2 located in library LIBX. Member MBR4 is created as a new member to file OUT1 in library LIB2 and contains the output from the default query. The record selection display is shown to allow you to specify which records from file FILE2 in library LIBx are output to The output contains detail records and summary records if any exist. Member MBR4 in file OUT1.

RVKACCAUT (Revoke Access Code Authority) Command

The Revoke Access Code Authority (RVKACCAUT) command takes away the access code authority for an individual user or a group of users.

Note: You can specify USER(*CURRENT) to remove access code authority from any access code you have authority to.

Restriction: You must have *ALLOBJ or *SECADMN special authority to take away access code authority for other users.



(1) The requester must have *ALLOBJ rights or *SECADM special authority to specify USER(*ALL).

(2) The requester must have *ALLOBJ rights or *SECADM special authority to specify any user profile other than his own.

Job: B,I Pgm: B,I Exec

ACC Parameter

Specifies the access code for which authority is revoked. The access code is a decimal number from 1 through 2047.

***ALL:** All access code authority for the user is revoked.

access-code: Specify a decimal number, from 1 through 2047, that specifies the access code authority that is revoked.

USER Parameter

Specifies the name of the user profile for whom access code authority is revoked.

***CURRENT:** Specifies that the access code authority of the user currently running this command is revoked. (This is the only valid option you can specify unless you have *ALLOBJ or *SECADM special authority.)

***ALL:** Access code authority is revoked from all users.

user-profile-name: Specify the name of the user profile that has access code authority revoked.

Examples

```
RVKACCAUT ACC(250) USER(*CURRENT)
```

This command takes away the access code authority of access code 250 from the user currently running this command.

```
RVKACCAUT ACC(300) USER(BILLY)
```

This command takes away the access code authority of access code 300 from user BILLY. This command was run by someone with *ALLOBJ or *SECADM special authority, or by BILLY. A user who runs this command for himself can enter USER(*CURRENT) or his own user profile name; they are the same.

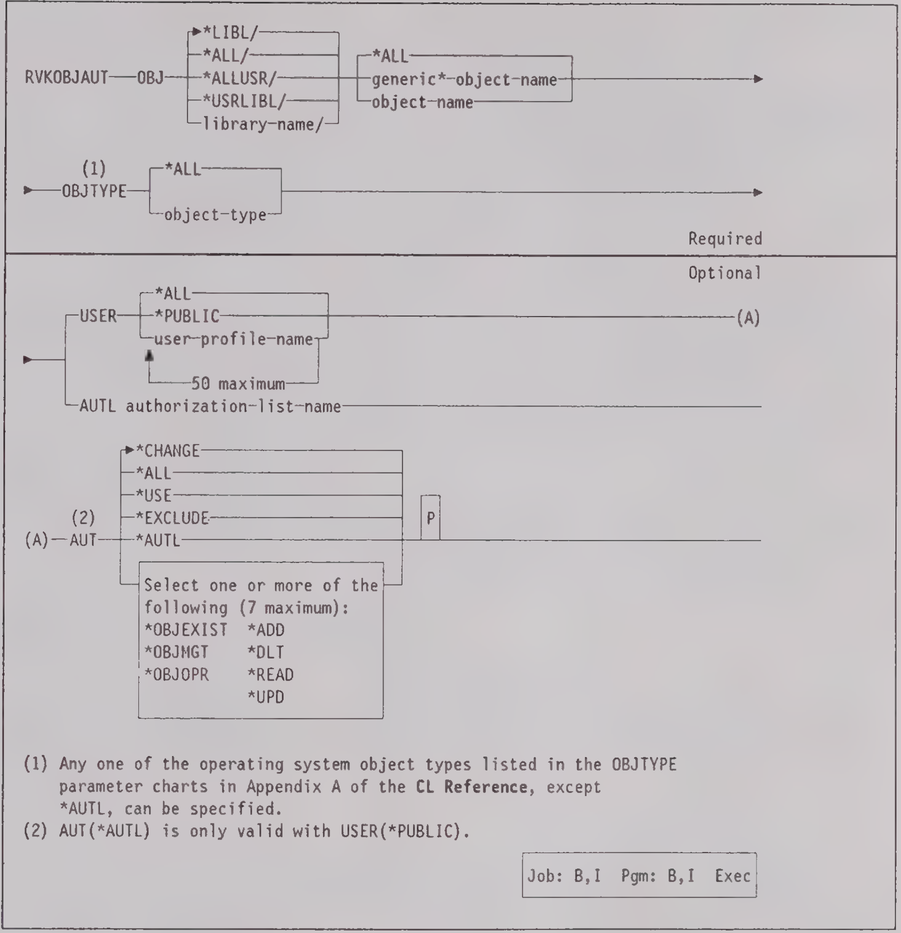
RVKOBJAUT (Revoke Object Authority) Command

The Revoke Object Authority (RVKOBJAUT) command is used to take away specific (or all) authority of use for the named objects from one or more users also named in the command or to remove the authority of an authorization list for the named objects. This command can be entered by the security officer, by an object's owner, or by a user who has object management authority for the object being removed. A user who has object management authority can remove only the explicit rights that he himself has. Also, a user may not be able to grant or remove rights for an object that has been allocated (locked) to another job. If revoking a specific (not *ALL) authority cannot be completely performed, a message is issued, which indicates the authorities that were not revoked.

Restrictions: (1) Before this command can be used to remove rights of use for a device, control unit, or line description, its associated device, control unit, or line must be varied on. Rights of use for a device cannot be revoked if a user is currently signed on to the device. Also, for display stations, if this command is not entered at the device for which rights are being revoked, it should be preceded by the Allocate Object (ALCOBJ) command and followed by the Delete Object (DLCOBJ) command. (2) Object types *DOC and *DOCL cannot be specified. You must use document interchange support. (3) Object type *AUTL cannot be specified. The Change Authorization List Entry (CHGAUTLE) or Remove Authorization List Entry (RMVAUTLE) commands must be used. (4) AUT (*AUTL) can be specified only with USER (*PUBLIC). (5) Only a user with *ALL authority or the owner can revoke the authorization list.

Security Risk

Revoking all authorities specifically given to a user for an object can result in the user having more authority than he did before the revoke. If a user has *USE authority for an object and *CHANGE on the authorization list that secures the object, revoking *USE authority results in the user having *CHANGE authority to the object.



(1) Any one of the operating system object types listed in the OBJTYPE parameter charts in Appendix A of the **CL Reference**, except *AUTL, can be specified.

(2) AUT(*AUTL) is only valid with USER(*PUBLIC).

Job: B,I Pgm: B,I Exec

OBJ Parameter

Specifies the name of the objects (library-name/object-name) to have specific authority revoked. Either a specific or a generic object name can be specified with a library name. If *ALL is specified, the name of a library must be specified. For more information on the use of generic functions, refer to Appendix G, "Generic Functions."

The possible library values are:

***LIBL:** The library list is searched.

***ALL:** All libraries, both system and user libraries, are searched.

***ALLUSR:** All user libraries are searched.

***USRLIBL:** All libraries on the user portion of the library list are searched.

library-name: Specify the name of the library that is searched.

***ALL:** All objects of the specified type (OBJTYPE) found in the search have specific rights revoked. You must specify the name of a library with *ALL.

generic-object-name:* Specify the generic name of the object that is to have specific rights revoked. A generic name can be specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic name and that the user has some authority for are prepared to remove their authorities.

object-name: Specify the name of the object that is to have specific rights revoked.

OBJTYPE Parameter

Specifies the object type of the object that has specific rights revoked. Any one of the operating system object types can be specified (except *AUTL); refer to the charts in the expanded description of the OBJTYPE parameter in Appendix A, "Expanded Parameter Descriptions" for a list of operating system object types.

***ALL:** All object types have specific authorities revoked.

object-type: Specify the type of the object (preceded by an asterisk) that is to have specific rights revoked.

USER Parameter

You cannot specify OBJTYPE(*ALL) with REFOBJTYPE(*OBJTYPE). Specifies the names of one or more users whose specific authorities to the named object are being removed away. If a user was granted the authority by USER(*PUBLIC) being specified in the Grant Object Authority (GRTOBJAUT) command, the same authorities can be revoked by *PUBLIC being specified in this parameter. If he was explicitly authorized (he was granted the authorities by having his name specified), his name must be specified in this parameter to remove the same authorities.

***ALL:** The authorities specified in the AUT parameter are being taken away from all enrolled users of the system except the owner, if they were publicly or explicitly authorized.

***PUBLIC:** The specified rights are taken away from users who do not have specific authority for the object, who are not on the authorization list, and whose group has no authority. Any users who have specific authority still retain their rights to the object.

user-profile-name: Specify the user names of one or more users that are having the specified authorities revoked. The authorities specified in the AUT parameter are being specifically taken away from each user specified. This parameter cannot be used to remove public authority from specific users; only authorities that were specifically granted to them can be specifically revoked.

***AUTL:** The public authority of the authorization list specified in the AUTL parameter is revoked for the object. The public authority for the object becomes *EXCLUDE.

authorization-list-name: Specify the name of the authorization list whose authority is removed from the object.

AUT Parameter

Specifies the authority you are taking away from the users.

***CHANGE:** Change authority allows the user to perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. In addition to the functions allowed with use (*USE) authority, the user can change the object. Change authority provides object operational authority and all the data authorities.

***ALL:** All authority allows the user to perform all operations on the object except those limited to the owner or controlled by authorization list management rights. In addition to the functions allowed with use (*USE) and change (*CHANGE), the user can control the object's existence and specify the security for the object. The user cannot transfer ownership of the object. If the object is an authorization list, the user cannot add, change, or remove users. Revoking *ALL authority from *PUBLIC causes the *PUBLICs authority to change to *EXCLUDE.

***USE:** Use authority allows the user to run a program or display the contents of a file. The user is prevented from changing the object. Use authority provides object operational authority and read authority.

***EXCLUDE:** Exclude authority prevents the user from accessing the object. *EXCLUDE cannot be revoked from *PUBLIC. An authority must be granted to *PUBLIC to change it from *EXCLUDE.

***AUTL:** Specifies that the authorization list is revoked from the object specified in the OBJ parameter. If public authority in the object is *AUTL, it is changed to *EXCLUDE. The authorization list's authority is then removed. Select up to seven of the following:

***OBJEXIST:** Object existence authority provides the authority to control the objects existence and ownership. These authorities are necessary for users who want to delete the object, free storage of the object, perform save and restore operations for the object or transfer ownership of an object. (If a user has special save system authority (*SAVSYS), he does not need object existence authority.) Object existence authority is required to create an object that has been named by an authority holder.

***OBJMGT:** Object management authority provides the authority to specify the security for the object, move or rename the object, and add members to data base files.

***OBJOPR:** Object operational authority provides authority to look at the description of an object and use the object as determined by the data authorities that the user has to the object.

The possible data authorities are:

***ADD:** Add authority provides the authority needed to add entries to an object; for example, adding job entries to a queue or adding records to a file.

***DLT:** Delete authority provides the authority needed to remove entries from an object; for example, removing messages from a message queue or records from a file.

***READ:** Read authority provides the authority needed to get the contents of an entry in an object or to run a program.

***UPD:** Update authority provides the authority needed to change the entries in an object.

Examples

```
RVKOBJAUT  OBJ(ARLIB/PROG1)  OBJTYPE(*PGM)  USER(*ALL)
```

This command removes the authorities (AUT was not specified; *CHANGE is assumed) from *all* users except the owner (who were either explicitly or publicly authorized) for the program (*PGM) named PROG1 located in the library named ARLIB.

```
RVKOBJAUT  OBJ(TSMITHPGM/MITHLIB)  +
  OBJTYPE(*PGM)  USER(TMSMITH)  +
  AUT(*OBJEXIST)
```

This command removes the object owner's (T M Smith) authority to delete a program (TSMITHPGM) in his library (SMITHLIB). (The object owner might do this to ensure that he does not delete the object by mistake.) If the owner ever wants to delete the object, he can grant himself object existence authority (using the Grant Object Authority (GRTOBJAUT) command) for the object.

```
RVKOBJAUT   OBJ(FILEX)  OBJTYPE(*FILE)  +
  USER(HEANDERSON)  AUT(*DLT *UPD)
```

This command removes delete and update authorities for the file named FILEX from the user H E Anderson.

```
RVKOBJAUT  OBJ(ARLIB/ARJOBD)  OBJTYPE(*JOBD)  +
  USER(RLJOHNSON)  AUT(*OBJEXIST)
```

The command removes the object existence authority for the object named ARJOBD from the user R L Johnson. ARJOBD is a job description that is located in the library named ARLIB.

```
RVKOBJAUT  OBJ(FILEX)  OBJTYPE(*FILE)  +
  AUTL(FILEUSERS)
```

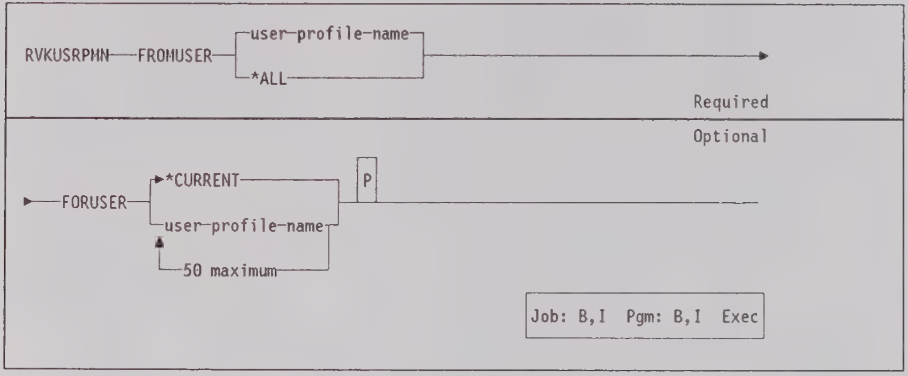
This command removes specific authorities for the file named FILEX from the users in the authorization list FILEUSERS.

RVKUSRPMN (Revoke User Permission) Command

The Revoke User Permission (RVKUSRPMN) command takes away user permission from one user (or all users) to access documents and folders on behalf of another user.

Note: If work is being done on behalf of another user at the time this command is run, any functions that started are completed; however, additional functions are not accepted.

Restriction: You must have *ALLOBJ or *SECADM special authority to revoke document authority for other users.



FROMUSER Parameter
Specifies the user profile name of the user who has permission revoked.

user-profile-name: Specify the name of the user profile that is no longer permitted to work on behalf of the user specified in the FORUSER parameter.

***ALL:** All users who are currently permitted to work on behalf of another user are no longer permitted.

FORUSER Parameter
Specifies the user profile name of the user on whose behalf the user specified in the FROMUSER parameter can no longer work.

***CURRENT:** Working on behalf of the user currently running this command is no longer permitted.

user-profile-name: Specify the name of the user profile on whose behalf other users are no longer permitted to work. You must have *ALLOBJ or *SECADM special authority to specify a user profile other than your own.

Example

```
RVKUSRPMN  FROMUSER(JOHNSON)  FORUSER(ANDERSON)
```

This command takes away user permission from JOHNSON for ANDERSON. The user JOHNSON is no longer allowed to work on behalf of ANDERSON.

SAVCHGOBJ (Save Changed Object) Command

The Save Changed Object (SAVCHGOBJ) command saves a copy of each changed object or group of objects located in the same library. When OBJ(*ALL) is specified, objects can be saved from up to 50 libraries. When saving to a save file, only one library can be specified. For data base files, only the changed members are saved. You can specify the date and time from which the system saves the objects or members.

Objects changed since the date and time specified are saved with the following exceptions:

- If OBJJRN(*NO) is specified, data base files currently being journaled are not saved, unless journaling was started after the specified date and time. This ensures that changes made to a physical file before journaling starts are not lost (because they were not recorded in a journal receiver).
- Freed objects (programs, files, journal receivers, and so forth) are not saved.
- Message (user-defined), job and output queue definitions, logical file definitions, save file descriptions, and data queue descriptions are saved, but the contents of those objects are not saved. Logical file access paths are saved if ACCPTH(*YES) is specified.

Specified objects are locked (shared, no update) while their change date and time are being checked, and the objects that were changed and the library where they reside remain locked for the duration of the save operation.

Saved objects can be restored with the Restore Object (RSTOBJ) command.

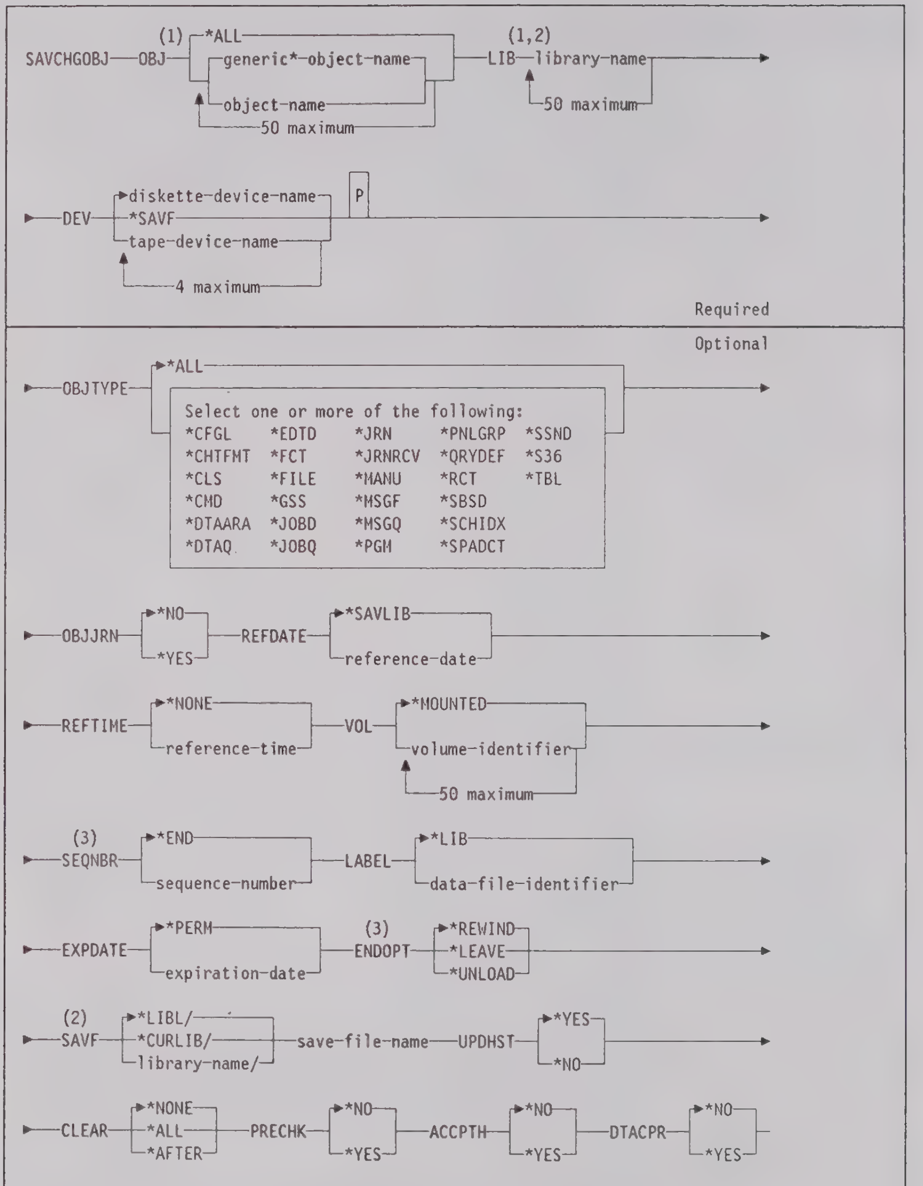
To determine the date and time that an object was changed, run the Display Object Description (DSPOBJD) command with DETAIL(*FULL) specified. For data base file members that were changed, run the Display File Description (DSPFD) command.

The types of objects that can be saved by this command are listed in the OBJTYPE parameter. The system saves the changed objects by writing a copy of each one on diskettes, tapes, or a save file. The description of each object is changed with the date, time, and place when it was last saved. (When saving to a save file, the UPDHST parameter controls this function.)

Note: This command ignores all file overrides currently in effect for the job.

Restrictions: To use this command, you must have the special authority *SAVSYS specified in your user profile by the SPCAUT parameter. Otherwise, you must have (1) object existence authority for each object specified, and (2) read authority for the specified library. (If you do not have the necessary authority to a specified object, all changed objects except that one are saved.) When saving to tape or diskette, you must have use authority to the device description. When saving to a save file, you must have add and use authority to the save file.

All diskettes used to save the changed objects must be initialized in the save/restore format. If tape is used, a standard labeled volume must be put on. Also, no changed object being saved can be changed by a job that is running at the time the save operation occurs.



(1) OBJ(*ALL) is required when more than one library is specified.
(2) Only one library can be specified when SAVF is used.
(3) Applies to tape devices only.

Job: B,I Pgm: B,I Exec

OBJ Parameter

Specifies the names of one or more changed objects to save. All the objects must be in the library specified in the LIB parameter. Specify the name of each object or the generic name of each group of objects to be saved.

If the OBJTYPE parameter is not specified, all the object types listed in the description of the OBJTYPE parameter are saved, provided they are in the specified library and have the specified names.

***ALL:** All changed objects in the specified libraries are saved, depending on the values specified for OBJTYPE.

generic-object-name:* Specify one or more generic names of groups of objects in the specified library whose changes are saved.

For more information or generic names, refer to Appendix G, "Generic Functions."

object-name: Specify as many as 50 names of specific changed objects to save. (Both generic names and specific names can be specified in the same command.)

LIB Parameter

Specify the name of the library that contains the changed objects to be saved. If *ALL is specified for the OBJ parameter, up to 50 library names may be specified.

Note: When multiple library names are specified, the overall save time is reduced, compared to saving objects from each of the libraries on individual commands. The best performance can be obtained by specifying the list of library names, in ascending order, by number of objects and/or members to be saved. For example, the first library saved should have the fewest number of objects and/or members to be saved.

Only one library can be specified when saving to a save file.

DEV Parameter

Specifies the name of the device on which the changed objects are saved. The device name must already be known on the system by a device description. If multiple devices are used, they must have compatible media formats.

diskette-device-name: Specify the name of the diskette device used to save the objects.

***SAVF:** The save or restore operation is done using the save file specified by the save file (SAVF) parameter.

tape-device-name: Specify the names of one or more tape devices that are used for the save operation. If you are using more than one tape device (up to a maximum of four), specify the names of the devices in the order you want them used. If you are using more than one tape volume, you may want to use more than one tape device, so that one tape volume can rewind and/or unload while another tape device is processing.

OBJTYPE Parameter

Specifies the types of OS/400 objects whose changes are saved. (Changes to *LIB, *USRPRF, *LIND, *CTLD, *DEVD, *MODD and *COSD objects are not looked at by the SAVCHGOBJ command for later display.) Data dictionaries and the associated files are not saved with the SAVCHGOBJ command.

***ALL:** Changes to all object types that are specified by name and are in the specified library are saved. If *ALL is also specified on the OBJ parameter, then changes to all objects in the libraries that are of the types shown in the following table are looked at.

object-type: Specify the value for each of the types of objects that are saved. The following types can be specified:

Value	Type of CPF Object
*CFGL	Configuration list
*CHTFMT	Chart format
*CLS	Class
*CMD	Command
*DTAARA	Data area
*DTAQ	Data queue
*EDTD	Edit description
*FCT	Forms control table
*FILE[1]	File
*GSS	Graphics symbol set
*JOBD	Job description
*JOBQ	Job queue
*JRN	Journal
*JRNRCV	Journal receiver
*MENU	Menu definition
*MSGF	Message file
*MSGQ	Message queue (user-defined only)
*OUTQ	Output queue
*PGM	Program
*PNLGRP	Panel group
*QRYDEF	Query definition
*RCT	Reference code translate table
*SBSD	Subsystem description
*SCHIDX	Search information search index
*SPADCT	Spelling aid dictionary
*SSND	Session description
*S36	S/36 Machine description
*TBL	Table

[1] For logical files and save files, only the definition is saved. To save the access path for a logical file, use the ACCPTH parameter. To save the contents of a save file, use the Save Save File Data (SAVSAVFDTA) command.

[2] The description of the queue is saved, but not the information contained in the queue.

OBJJRN Parameter

Specifies if changes to objects currently being journaled (as specified in the Start Journal Physical File (STRJRNPF) command) are saved.

***NO:** Changes to journaled objects (for which journaling was started before the date and time specified in the REFDATE and REFTIME parameters) are not saved, but the journal receivers are saved. If the STRJRNPF command was run after the specified date and time, all changed members are saved. The date and time of the last journal start can be displayed by the Display File Description (DSPFD) command.

***YES:** Changes to journaled objects are saved.

REFDATE Parameter

Specifies the date when changed objects in the specified libraries are saved.

***SAVLIB:** The changed objects are saved, starting from the date of the last running of the Save Library (SAVLIB) command. If the specified library was never saved, a message is issued and the operation ends.

reference-date: Specify the date when changed objects are to be saved. If this date is later than the date of the running of this command, a message is issued and the operation ends.

REFTIME Parameter

Specifies a time when the changed objects of the specified library are to be saved.

***NONE:** No explicit time is specified. The changed objects are saved from the start of the date specified by the REFDATE parameter.

reference-time: Specify the time when changed objects are to be saved. If REFDATE(*SAVLIB) is specified, no reference time can be specified. If this time is greater than the time of the running of this command, a message is issued and the operation ends.

VOL Parameter

Specifies the volume identifiers of the tape volumes or diskette volumes on which the object data is to be saved. The volumes must be put on in the same order as the volume identifiers are specified in this parameter. A maximum of 50 entries can be specified. (For more information on VOL parameter, refer to Appendix A, "Expanded Parameter Descriptions.")

***MOUNTED:** The volume that is currently on the devices specified by the DEV parameter is used.

volume-identifier: Specify the identifiers of one or more volumes in the order you want them put on and used to save the objects.

SEQNBR Parameter

Specifies, only when tape is used, which sequence number is used as the starting point for saving changed objects.

***END:** The system saves the objects after the last sequence number on the tape volume.

sequence-number: Specify the sequence number of the file used as a starting point to save changed objects.

LABEL Parameter

Specifies the name that identifies the data file on the tape or diskette that is to be used for the save. If the LABEL parameter is used on the save command, the same label must be specified on the restore command.

***LIB:** The file label is created by the system using the name of the library specified on the LIB parameter.

data-file-identifier: Specify the data file identifier (a maximum of 17 characters) of the data file used for the save. This option is valid only for the single-library saves.

EXPDATE Parameter

Specifies the end date of the save/restore file. If a date is specified, the save/restore file is protected and cannot be overwritten until the specified end date. This parameter is valid for both tape and diskette.

***PERM:** The save/restore file is protected permanently.

expiration-date: Specify the end date when the save/restore file ends.

Note: For operations to diskette, the expiration date specified must be later than the date of the save operation. Otherwise, the save/restore files that have ended may be lost when the next save/restore file is written during the save operation.

ENDOPT Parameter

Specifies, only when tape is used, what operation is automatically done on the tape volume after the save operation ends. If more than one reel is included, this parameter applies only to the last reel used, all other reels are rewound and unloaded when the end of the tape is reached.

Note: If no objects are saved (for example, no objects meet the selection value specified on the command), the tape volume is not opened and the ENDOPT parameter is ignored.

***REWIND:** The tape is automatically rewound, but not unloaded, after the save operation has ended.

***LEAVE:** The tape is not rewound; another save operation can start at the current position on the tape.

***UNLOAD:** The tape is automatically rewound and unloaded after the save operation has ended.

SAVF Parameter

Specifies the qualified name of the save file that should be used to contain the saved data. The save file should be empty, unless CLEAR(*ALL) is specified.

The possible library values are:

***LIBL:** The library list is used to locate the save file.

***CURLIB:** The current library for the job is used to locate the save file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the save file is located.

UPDHST Parameter

Specifies if the save/restore history information of each object saved is changed with the date, time, and location of this save operation. The save history information for an object is available using the Display Object Description (DSPOBJD) command. This information is used to determine which journal entries are processed when RCVRNG(*LASTSAVE) and FROMENT(*LASTSAVE) are used on the Apply Journaled Changes (APYJRNCHG) command.

Note: This parameter applies only when saving to a save file (SAVF parameter).

***YES:** The last save date, time, and location information should be updated in each object saved. This value works the same way as a save operation to diskette or tape.

***NO:** The save history information contained in the description of each object saved should not be updated.

Note: UPDHST(*NO) should only be used for a save operation that is not intended for recovery or saving. For example, if the save data is transmitted, record by record, to another system and the save file then deleted, you probably do not want to update the save history information.

CLEAR Parameter

Specifies if an uncleared save file, or uncleared diskettes or tapes encountered during the save operation are not automatically cleared. An uncleared diskette or tape is one containing a file with an expiration date later than the date of the save operation (which includes files protected permanently using EXPDATE(*PERM)). (This parameter does not control initializing the diskettes; diskettes should already be initialized in the save/restore format; tapes should already be standard labeled.)

***NONE:** None of the uncleared tapes, diskettes, or save files encountered during the save operation are automatically cleared. If the save operation cannot proceed because an uncleared diskette, tape, or save file is encountered, an inquiry message is sent to the operator, allowing him to end the save operation or to specify that the currently selected diskette, tape, or save file be cleared so the operation can continue. If a save file is not cleared, the inquiry message is sent to the work station message queue for an interactive job, or to the operator for a batch job. The save file should be empty, or all diskettes or tapes used to perform the save operation should be cleared before the save command is issued.

***ALL:** All the uncleared diskettes, tapes, or save files encountered during the save operation are automatically cleared. If tapes are used and a sequence number is specified, the tape is cleared and, starting with that sequence number, all tapes following the first tape are cleared.

***AFTER:** All the uncleared diskettes or tapes after the initial diskette or tape are automatically cleared. This option is not valid for save/restore operations to save files. If the operation cannot proceed because the first diskette or tape is uncleared, an inquiry message is sent to the system operator, allowing him to end the operation or to specify that the currently selected diskette or tape be cleared so the operation can continue. All diskettes or tapes used to perform the save operation should be cleared before the save command is issued.

PRECHK Parameter

Specifies if the save operation for a library should end if all objects specified by this command do not satisfy the following conditions of the save operation: the objects exist, they were not previously found to be damaged, they are not locked by another job, and the requester of the save operation has authority to save the objects.

***NO:** Allows the save operation for a library to continue, saving only those objects that can be saved.

***YES:** Ends the save operation for a library before any data is written, if, after all specified objects are checked, one or more objects cannot be saved. If multiple libraries are specified, the save operation continues with the next library.

ACCPTH Parameter

Specifies if the logical file access paths dependent on the physical files being saved are also saved. The access paths are saved only if all members on which the access paths are built are included in this save operation. The access paths must be MAINT(*IMMED) or MAINT(*DLY), and the save operation must be able to get a *SHRNUP lock on the logical file containing the access path. If any of these conditions are not met, the access path is not saved, but the save of the physical files proceeds normally. Informational messages are sent indicating the number of logical file access paths saved with each physical file. The physical files on which the access path is built must exist in the same library. This parameter does not save logical file objects; it only controls the saving of the access paths. If logical files are also saved in the same command, the saving of access paths is still determined by this parameter. More information on the restoring of saved access paths is in the *Data Base Guide*.

***NO:** Only those objects specified on the command are saved. No logical file access paths are saved.

***YES:** The specified physical file(s) and all eligible logical file access paths over them are saved.

DTACPR Parameter

Specifies if data compression is used.

***NO:** No data compression is performed.

***YES:** Program data compression is performed.

Example

```
SAVCHGOBJ   OBJ(ORD*)  LIB(DSTPRODLB)  DEV(TAP1)  OBJTYPE(*FILE)  +
  REFDATE(122288)
```

This command saves all changed files with names that start with the characters ORD in library DSTPRODLB that were changed since the 22nd of December, 1988.

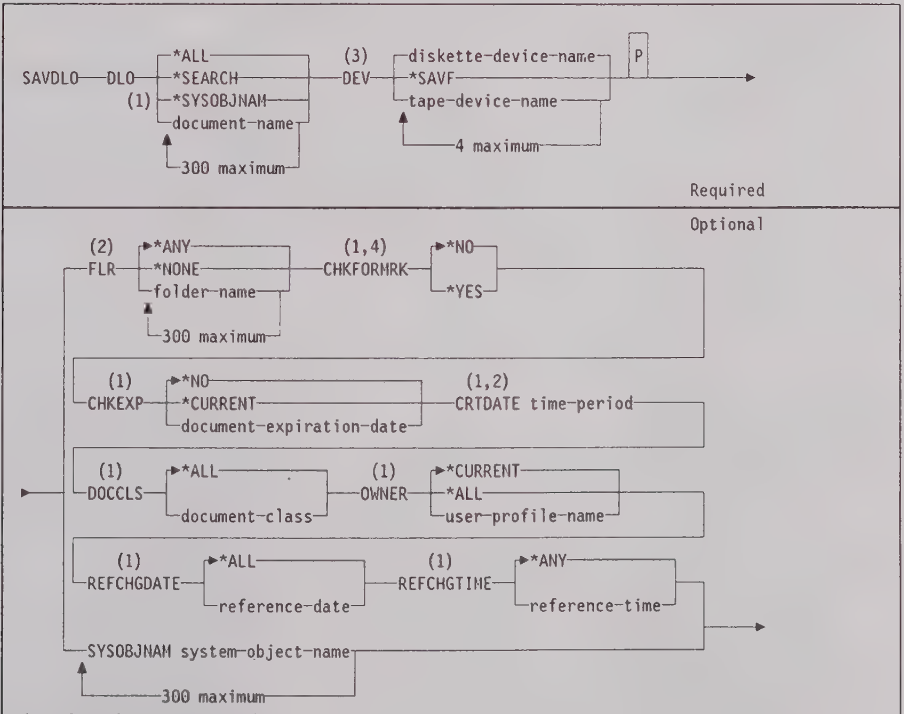
SAVDLO (Save Document Library Object) Command

The Save Document Library Object (SAVDLO) command saves a copy of the specified documents or folders located in library QDOC. The system saves the documents or folders on offline storage by writing a copy of each one on diskette or tape or to an online save file. The documents or folders are not affected in the system unless the command specifies that the storage is to be freed or deleted.

Notes:

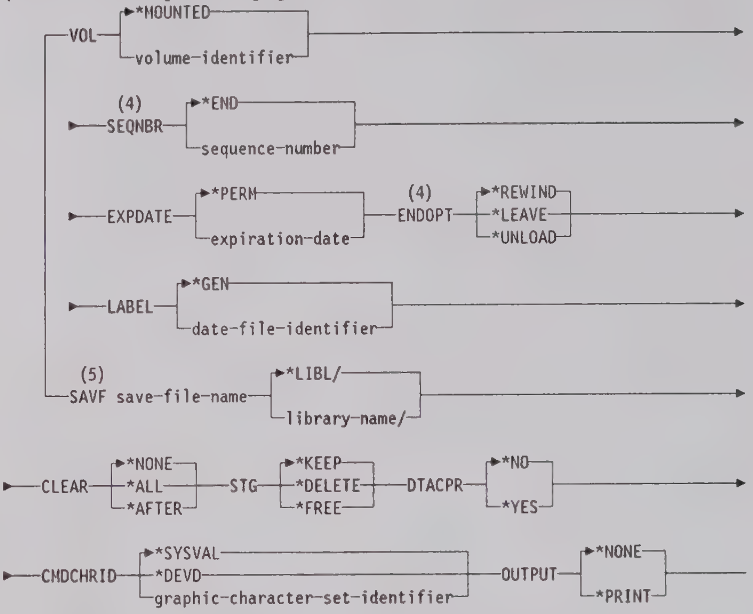
1. If library QDOC is not found, the save operation ends before any documents or folders have been saved, and a message is sent.
2. Determining of document or folder ownership does not include checking group profiles if one is associated with the specified user profile.

Restrictions: (1) To use this command, you must have either the *ALLOBJ or *SAVSYS special authority or you must have *ALL authority for each saved document or folder. (2) All object authority or *SAVSYS special authority is required if a search value is specified and the owner is not equal to the user making this request.



(continued on next page)

(continued from previous page)



(1) CHKFORMRK, CHKEXP, CRTDATE, CLS, REFCHGDATE, REFCHGTIME, and OWNER are allowed if DOC(*SEARCH) is specified.
(2) CRTDATE contains 2 lists of 2 elements each.
(3) Applies to diskette device only.
(4) Applies to tape device only.
(5) Applies to online save file only.

Job: B,I Pgm: B,I Exec

DLO Parameter

Specifies the documents and folders to save. To save a folder, DLO(*ALL) must be specified. All other DLO parameter values save documents only.

***ALL:** All documents or folders, all folderless documents, or all documents and folders in the document saved.

***SEARCH:** All documents that meet the specified search value are saved. Search value is specified using the following parameters: FLR, CHKFORMRK, CHKEXP, CRTDATE, CLS, OWNER, REFCHGDATE, and REFCHGTIME.

***SYSOBJNAM:** The documents with the system object names specified in the SYSOBJNAM parameter are saved.

document-name: Specify the user-assigned name of the documents that are saved. A maximum of 12 characters can be specified.

DEV Parameter

Specifies the type of storage media the documents and/or folders are saved to.

diskette-device-name: Specify the name of the documents and/or folders to save to diskette files.

***SAVF:** The documents and/or folders are saved to the save file specified in the SAVF parameter.

tape-device-name: Specify the name of the documents and/or folders to save to tape. A maximum of four tape devices can be specified.

FLR Parameter

Specifies the folder in which the object specified in the DLO parameter is located. This parameter is ignored if DLO(*SYSOBJNAM) is specified.

***ANY:** A folder value is not specified if DLO(*SEARCH) is specified. All documents in the document library are saved if DLO(*ALL) is specified. FLR(*ANY) is not valid if DLO(document-name) is specified.

***NONE:** The documents saved are documents not in any folder. FLR(*NONE) is valid only if DLO(*ALL) or DLO(*SEARCH) is also specified.

folder-name: Specify the user-assigned name of the folder that the documents are saved from.

CHKFORMRK Parameter

Specifies if documents that have been marked for offline storage are saved. This parameter is valid only for documents.

***NO:** All documents that meet the other search values for this save operation are saved regardless of whether they were marked for offline storage or not.

***YES:** Only those documents that meet the other search values and are marked for offline storage are saved.

CHKEXP Parameter

Specifies the date for which all documents with an expiration date less than this date are saved. This date is assigned by the user when the document was created to specify when the document is no longer needed.

***NO:** The expiration date is ignored.

***CURRENT:** All documents with an expiration date less than today's date are saved.

document-expiration-date: Specify a date for all documents that have an expiration date less than the date it is saved.

CRTDATE Parameter

Specifies that documents created during the time period specified are to be saved. If a document is to be selected, it must be created on or after the starting time and date and on or before the ending time or date. This parameter contains two lists of two elements, each representing the starting time and date and the ending time and date as follows:

```
((start-time  start-date) (end-time  end-date))
```

If CRTDATE is not specified, the following values are assumed:

```
((*AVAIL  *CURRENT) (*AVAIL  *END))
```

Starting Time: Use one of the following to specify the starting time. Documents must have been created at or after this time to be selected. Documents created before this time are not selected.

***AVAIL:** Documents created at any time are eligible for selection.

start-time: Specify to save only those documents that were created at or after the specified time on the starting date. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. If colons are used to separate the time values, the character string must be enclosed in apostrophes ('hh:mm:ss'). When the start-time is used as a search value, the start-date must not be *BEGIN.

Starting Date: Use one of the following to specify the selected starting date. Documents must have been created on or after this date to be selected. Documents created before this date are not selected.

***CURRENT:** Save only those documents created on today's date, after the starting time (if specified).

***BEGIN:** Documents created on any date can be selected. (The starting date and time is ignored.)

start-date: Specify to save only those documents created on or after the date specified here. The date must be specified in the format specified by the system value QDATFMT and, if separators are used, QDATSEP.

Ending Time: Use one of the following to specify the selected ending time. Documents must have been created at or before this time to be selected. Any documents created after the specified time are not selected.

***AVAIL:** Documents created at any time are eligible for selection.

end-time: Save only those documents that were created at or before the specified time on the ending date. The time can be specified as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. If colons are used to separate the time values, the character string must be enclosed in apostrophes ('hh:mm:ss'). When the end-time is to be used as a search value, the end-date must not be *END.

Ending Date: Use one of the following to specify the ending date. Documents must have been created on or before this date to be selected. Documents created after this date are not selected.

***END:** Documents created on any date are eligible for selection. The ending time is ignored when *END is specified.

end-date: Documents created on or before this date are selected. The date must be specified in the format specified by the system value QDATFMT and, if separators are used, QDATSEP.

DOCCLS Parameter

Specifies the class of documents that is saved. The class is assigned by the user when the document is created. This parameter is valid for documents only.

***ALL:** The document class is not used to select documents being saved.

document-class: Specify a character string, 1 through 16 characters in length, that is used to select documents being saved.

OWNER Parameter

Specifies the owner of the documents that are being saved.

***CURRENT:** Documents owned by the user issuing this command are saved.

***ALL:** The OWNER parameter is not used to select documents to be saved. *ALLOBJ or *SAVSYS special authority is required if OWNER(*ALL) is specified.

user-profile-name: Specify the name of the user who owns the documents that are being saved. All documents owned by this user are saved. *ALLOBJ or *SAVSYS special authority is required if a user profile is selected other than the user profile of the user issuing this command.

REFCHGDATE Parameter
Specifies that the documents that have changed after the date specified are saved. The change date is updated when the document content or description is changed. This parameter is valid for documents only.

***ANY:** No specific date is specified. Documents are saved regardless of the date they were last changed.

reference-date: Specify the date after which the changed documents are saved.

REFCHGTIME Parameter
Specifies that documents that have changed after the specified time are saved. The change time is updated when the document content or description is changed. This parameter is valid for documents only.

***ANY:** No specific time is specified. The documents are saved regardless of the time they were changed.

reference-time: Specify the time after which changed documents are saved. A reference date must be specified if reference time is specified.

SYSOBJNAM Parameter
Specifies the system object name of the document that is saved. A full 10 characters must be specified. A maximum of 300 names can be specified. If DLO(*SYSOBJNAM) is not specified, the SYSOBJNAM parameter is ignored.

VOL Parameter
Specifies the volume identifiers of the tape volumes (or diskette volumes) on which the documents or folders are saved. The volumes must be in the same order as the volume identifiers are specified in the parameter. (For an expanded description of the VOL parameter, see Appendix A, "Expanded Parameter Descriptions.")

***MOUNTED:** The documents or folders that are saved are copied on whatever volumes are on the device.

volume-identifier: Specify the identifiers of one or more volumes in the order in which they are on the device and used to save the documents or folders. For tape volumes and single-diskette volumes, a maximum of 6 characters identify each volume.

SEQNBR Parameter
Specifies the starting tape sequence number to use for saving the documents or folders.

***END:** The system saves documents or folders after the last sequence number on the tape volume.

sequence-number: Specify the sequence number of the file that is used as a starting point to save the documents or folders.

EXPDATE Parameter
Specifies the date that the save/restore files are no longer needed. If a date is specified, the save/restore files are protected and cannot be overwritten until the specified date that they are no longer needed. This parameter is valid for both tape and diskette.

***PERM:** The save/restore files are protected permanently.

expiration-date: Specify the date that the save/restore files are no longer needed.

Note: For operations to diskette, the date specified as no longer needed must be later than the date of the save operation; otherwise, the save/restore files may be lost when the next save/restore file is written during the save operation.

ENDOPT Parameter
Specifies, only when tape is used, what operation is automatically performed on the tape volume after the save operation ends. If more than one reel is used, this parameter applies only to the last reel used; all other reels are rewound and unloaded when the end of the tape is reached.

***REWIND:** The tape is automatically rewound, but not unloaded, after the save operation has ended.

***LEAVE:** The tape is not rewound; another save operation can start at the current position on the tape.

***UNLOAD:** The tape is automatically rewound and unloaded after the save operation has ended.

LABEL Parameter
Specifies the file label that is given to the file that is written on the media during the save operation. This parameter is only valid for tape and diskette and for saving less than 8000 documents or folders.

***GEN:** The file label is created by the system. If folders containing more than 8000 objects are saved, and the user has specified a file label, the file labels have the form DLO001, DLO002, and so forth. For all other documents and folders saved, the file labels created have the same form as the labels generated by the LABEL parameter on the Save Object (SAVOBJ) command. For example, since the document library is QDOC, generated labels have the form of QDOC.0001, QDOC.0002, and so forth.

date-file-identifier: Specify the date file identifier that is used as the label for the data file being saved. A maximum of 17 characters can be specified. This option is not valid when folders containing more than 8000 objects are saved.

SAVF Parameter

Specifies the name of the save file.

The possible library values are:

LIBL: The library list is used to locate the save file.

library-name: Specify the library in which the save file is located.

CLEAR Parameter

Specifies if uncleared diskettes or tapes or online save files encountered during the save operation are automatically cleared. (This parameter does not control initializing the diskettes or tapes. Diskettes should already be initialized in the save/restore format; tapes should be standard labeled.)

***NONE:** Uncleared diskettes or tapes encountered during the save operation are not automatically cleared. For the first diskette or tape used, the save operation starts at the end of the data already there, if enough space is available. If enough space is not available to start the save operation on that volume, a message is sent and the operation ends. If, after the first diskette or tape is used, any uncleared diskette or tape is found, an inquiry message is sent to the operator, allowing him to determine the save operation or to specify that the currently selected diskette or tape be cleared so the operation can continue.

***ALL:** Uncleared diskettes or tapes found during the save operation are automatically cleared so the save can continue. If tapes are used and a sequence number is specified, the tape cleared starting with that sequence number; all tapes following the first tape are cleared.

***AFTER:** All uncleared diskette or tapes after the initial diskette or tape are automatically cleared.

STG Parameter

Specifies if the system storage that is occupied by the document being saved is kept, deleted, or freed after the save operation is completed. This parameter applies only to documents.

***KEEP:** The storage occupied by the document is kept. The document is still available on the system after the save.

***DELETE:** The document object and all search terms are deleted from the system after the save. This includes both the data portion of the document and the document description. Documents that are marked for saving with delete are not deleted unless CHKFORMRK(*YES) is specified. A diagnostic message is sent for each document that is saved but not deleted.

***FREE:** The search terms maintained for the document and the document's description are kept. The document can still be identified on a DIA SEARCH result. The document is not retrieved until the document is restored to the system. Documents that are marked for saving are not freed unless CHKFORMRK(*YES) is specified. A diagnostic message is sent for each document that is saved but not freed.

DTACPR Parameter

Specifies if data compression is used.

***NO:** No data compression is done.

***YES:** Program compression is performed.

CMDCHRID Parameter

Specifies the character identifier (graphic character set and code page) for the data being entered as command parameter values (applies to CLS parameter). The character identifier is related to the display device that was used to enter the command. More information about CHRID processing is in the *Data Management Guide.*

***SYSVAL:** The system determines the graphic character set and code page values for the command parameter from the QCHRID system value.

***DEVD:** The system determines the graphic character set and code page values for the command parameter from the display device description where this command was entered. This option is valid only when specified from an interactive job. If specified in an interactive CL program or a batch job, an error occurs.

graphic-character-set-identifier: Specify the graphic character set and values that were used to create the command parameter. Each value can be up to 3 bytes in length and must be separated by at least one blank.

OUTPUT Parameter

Specifies if a list with the information about the saved documents or folders is created. This list shows information from the Interchange Document Profile (IDP) of each saved document.

***NONE:** No list is printed.

***PRINT:** The list is printed with the job's spooled output on a printer.

Examples

```
SAVDLO  DLO(DOCX)  FLR(FOLDERA) DEV(QDKT5) STG(*FREE)
```

This command saves the document named DOCX in folder FOLDERA to the diskette file DKT5. As part of the save operation, the system storage that was occupied by the data portion of the document is freed.

```
SAVDLO  DLO(*SEARCH)  DEV(QTAP5)  CLS(MEMOS)  +
  VOL(TOM)  CLEAR(*ALL)
```

This command saves all documents with a document class of MEMOS on the tape file TAP5; the tape must have a volume identifier of TOM. If the tape has not been cleared, it is cleared automatically when documents are saved.

```
SAVDLO DLO(*ALL)  DEV(QDKT3)
```

This command saves all documents and folders in the document library to diskette file named DKT3.

```
SAVDLO  DLO(*SYSOBJNAM)  DEV(*SAVF)  +
  SYSOBJNAM(HZ83B55219)  SAVF(SAVF1)
```

This command saves the document with the system object name HZ83B55219 to a save file SAVF1. The storage occupied by the document is not freed because STG(*KEEP) is assumed.

```
SAVDLO  DLO(*ALL) DEV(*SAVF) +
   FLR(BILL TOM MARY) +
   SAVF(SAVFIL1)
```

This command save the folders BILL, TOM, and MARY, and all the documents in these folders.

```
SAVDLO  DLO(*ALL) DEV(QTAP5)  +
   REFCHGDATE('01/01/88')
```

This command saves all documents changed after 01/01/88. This command is useful for saving changes between backups of the documents. This command is similar to the Save Changed Objects (SAVCHGOBJ) used for other object types.

SAVLIB (Save Library) Command

The Save Library (SAVLIB) command saves a copy of 1 to 50 specified libraries, or all libraries, except for QSYS, QDOC, and other OS/400-required libraries. The QGPL library and other IBM-supplied libraries and all user-created libraries can be saved. When saving to a save file, only one library can be specified. To save the specified libraries on offline storage, a copy of each library is written on diskettes or tapes.

Documents contained in QDOC library can be saved by the Save Document Library Object (SAVDLO) command.

The SAVLIB command saves the entire library; this includes the library description, the object descriptions, and the contents of the objects in the library. For job queues, message queues, output queues, data queues, logical files, and save files only the object definitions are saved, not the contents. (Logical file access paths may be saved, however, by use of the ACCPTH parameter. The contents of a save file can be saved by use of the Save Save File Data (SAVSAVFDTA) command.) It also includes program temporary fix (PTF) status information for all PTFs in the library. The libraries and their objects are not affected in the system unless the command specifies that the storage is to be freed. However, the description of each library and each object is updated with the date, place, and time when it was last saved (unless UPDHST(*NO) is specified when saving to a save file). If all user libraries are saved, the date, time, and place are updated in the history information for the data area QSAVLIBALL in QSYS.

The types of objects saved by this command are the same as those listed in the OBJTYPE parameter of the Save Object (SAVOBJ) command. Certain OS/400 objects that are not contained in user libraries (such as user profiles) are not saved by this command. They can be saved by the Save System (SAVSYS) command.

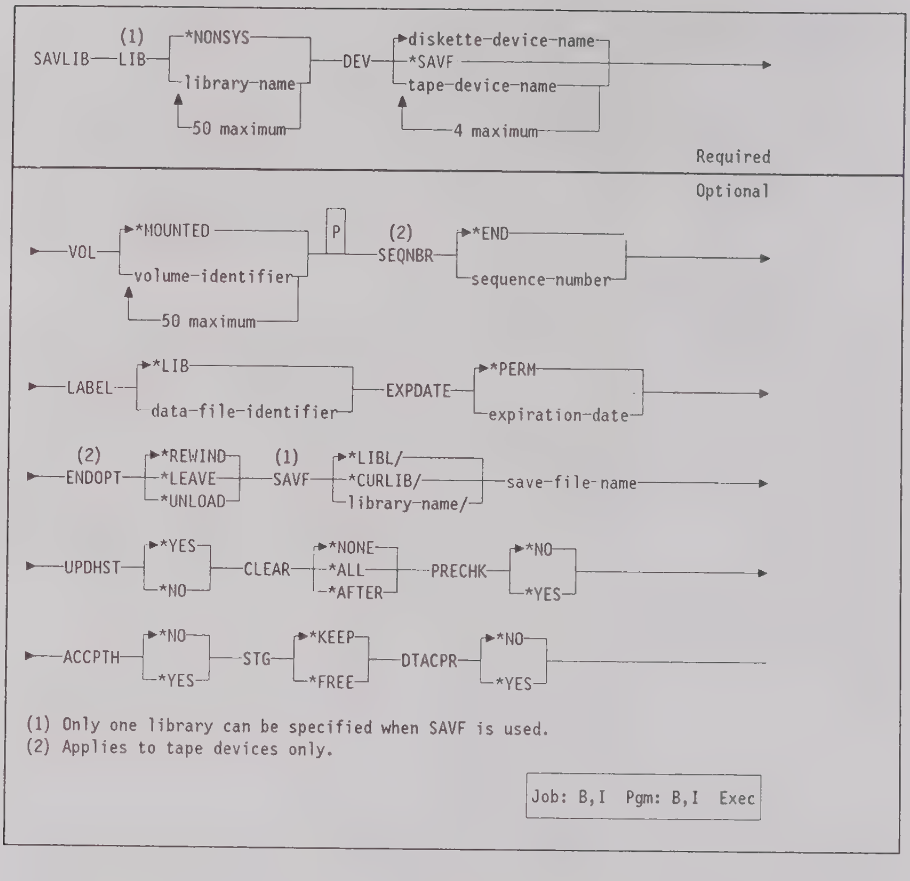
Note: This command ignores all file overrides that are currently in effect for the job.

Restrictions: To use this command, you must have either the special authority *SAVSYS specified in your user profile by the SPCAUT parameter, or you must have (1) read authority for, or be the owner of, each library specified and (2) object existence authority for each object in the library. (If you do not have the correct authorities for all of the libraries and objects specified, only those for which you do have the authorities are saved.)

When saving to tape or diskette, you must have use authority to the device description. When saving to a save file, you must have add and use authority to the save file.

All diskettes used to save the libraries should be initialized in the save/restore format. If tape is used, a standard labeled tape volume should be put on. Also, no library being saved, or the objects in it, can be updated by a job that is running at the time the save operation occurs.

If a library being saved contains more than 8000 internal objects, the save operation ends for that library before any objects are saved, and an error message is sent.


SAVLIB—LIB (1) *NONSYS / library-name, 50 maximum
DEV diskette-device-name / *SAVF / tape-device-name, 4 maximum
Required
Optional
VOL *MOUNTED / volume-identifier, 50 maximum
P
SEQNBR (2) *END / sequence-number
LABEL *LIB / data-file-identifier
EXPDATE *PERM / expiration-date
ENDOPT (2) *REWIND / *LEAVE / *UNLOAD
SAVF (1) *LIBL/ / *CURLIB/ / library-name/ save-file-name
UPDHST *YES / *NO
CLEAR *NONE / *ALL / *AFTER
PRECHK *NO / *YES
ACCPTH *NO / *YES
STG *KEEP / *FREE
DTACPR *NO / *YES
(1) Only one library can be specified when SAVF is used.
(2) Applies to tape devices only.
Job: B,I Pgm: B,I Exec

LIB Parameter

Specifies which libraries, or all user libraries, are saved on offline storage.

***NONSYS:** Specifies that all user-created libraries, the QGPL library, and licensed program libraries, such as QRPG and QIDU, are saved. All subsystems must be ended by the End Subsystem (ENDSBS) (with SBS(*ALL) specified) or End System (ENDSYS) command before this option is specified. When *NONSYS is specified, the user libraries are saved in alphabetical order on the media. More information on restoring data base files that have dependencies that cross library boundaries is in the *Data Base Guide*.

library-name: Specify the names of 1 to 50 libraries to be saved. The libraries QSYS, QSRV, QTEMP, QSPL, QDOC, and QRECOVERY cannot be specified.

Notes:

1. When multiple library names are specified, the overall save time is reduced, compared to saving objects from each of the libraries on individual commands. The best performance can be obtained by specifying the list of library names, in ascending order, by number of objects and/or members to be saved. For example, the first library saved should have the fewest number of objects and/or members to be saved.
2. Only one library can be specified when saving to a save file.

DEV Parameter

Specifies the name of the device on which the libraries are saved. The device name must already be known on the system by a device description.

diskette-device-name: Specify the name of the diskette device that is used to save or restore the objects.

***SAVF:** The save or restore operation is done using the save file specified by the save file (SAVF) parameter.

tape-device-name: Specify the name of one or more tape devices used for the save or restore. If you are using more than one tape device (up to a maximum of four), specify the names of the devices in the order you want them used. If you are using more than one tape volume, you may want to use more than one tape device so that one tape volume can rewind and/or unload while another tape device is processing.

VOL Parameter

Specifies the volume identifiers of the tape volumes or diskette volumes on which the library data is saved. The volumes must be put on in the same order as the volume identifiers are specified in this parameter. A maximum of 50 entries can be specified. (More information on the VOL parameter is in Appendix A, "Expanded Parameter Descriptions.")

***MOUNTED:** The libraries are saved on whatever volumes are put on the device.

volume-identifier: Specify the identifiers of one or more volumes in the order they are on the device and used to save the libraries.

SEQNBR Parameter

Specifies, only when tape is used, which sequence number is used as the starting point for the save/restore process.

***END:** The system saves the file to the sequence number after the last sequence number on the tape.

sequence-number: Specify the sequence number of the file used as a starting point to save the library files.

LABEL Parameter:

Specifies the name that identifies the data file on the tape or diskette that is used for the save. If the LABEL parameter is used on the save command, you must specify this label on the restore command.

***LIB:** The file label is created by the system using the name of the library specified on the LIB parameter.

data-file-identifier: Specify the date file identifier (a maximum of 17 characters) if the data file is used for the save or restore. This option is valid only for single library restores.

EXPDATE Parameter

Specifies the end date of the save/restore file. If a date is specified, the save/restore file is protected and cannot be overwritten until the specified expiration date. This parameter is valid for both tape and diskette.

***PERM:** The save/restore file is protected permanently.

expiration-date: Specify the date when the save/restore file ends.

Note: For operations to diskette, the end date specified must be later than the date of the save operation. Otherwise, the ended save/restore files may be lost when the next save/restore file is written during the save operation.

ENDOPT Parameter

Specifies, only when tape is used, what operation is automatically done on the tape volume after the save operation ends. If more than one volume is included, this parameter applies only to the last volume used; all other volumes are rewound and unloaded when the end of the tape is reached.

***REWIND:** The tape is automatically rewound, but not unloaded, after the save operation has ended.

***LEAVE:** The tape is not rewound; another save operation can start at the current position on the tape.

***UNLOAD:** The tape is automatically rewound and unloaded after the save operation has ended.

SAVF Parameter

Specifies the qualified name of the save file that is used to contain the save data. The save file should be empty, unless CLEAR(*ALL) is specified.

The possible library values are:

***LIBL:** The library list is used to locate the save file.

***CURLIB:** The current library for the job is used to locate the save file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the save file is located.

UPDHST Parameter

Specifies if the save/restore history information of each object saved is to be updated with the date, time, and location of the save operation. The save history information for an object is available using the Display Object Description (DSPOBJD) command. The save history information is used to determine which journal entries are processed when RCVRNG(*LASTSAVE) and FROMENT(*LASTSAVE) are used on the Apply Journaled Changes (APYJRNCHG) command.

Note: This parameter only applies when saving to a save file (SAVF parameter).

***YES:** The last save date, time, and location information is updated for each object saved. This value works the same way as a save operation to diskette or tape.

***NO:** The save history information contained in the description of each object saved should not be updated.

Note: UPDHST(*NO) should only be used for a save operation that is not intended for recovery or saving. For example, if the save data is transmitted, record by record, to another system and the save file immediately deleted, you probably will not want to update the save history information.

CLEAR Parameter

Specifies if an uncleared save file, or uncleared diskettes or tapes encountered during the save operation are automatically cleared. An uncleared diskette or tape is one containing a file with an expiration date later than the date of the save operation (which includes files protected permanently using EXPDATE(*PERM)). This parameter does not control initializing diskettes or tapes; diskettes should already be initialized in the save/restore format and tapes should be standard labeled. If a diskette is not initialized, is incorrectly named, or a tape volume that is not initialized is encountered during the save operation, an inquiry message allows the volume to be initialized or renamed.

***NONE:** None of the uncleared tapes, diskettes, or save files encountered during the save operation are automatically cleared. If the save operation cannot proceed because an uncleared diskette, tape, or save file is encountered, an inquiry message is sent to the operator, allowing him to end the save operation, or to specify that the currently selected diskette, tape, or save file be cleared so the operation can continue. If a save file is not cleared, the inquiry message is sent to the work station message queue for an interactive job, or to the operator for a batch job. The save file should be empty, or all diskettes or tapes used to perform the save operation should be cleared before the save command is issued.

***ALL:** All uncleared diskettes, tapes, or save files encountered during the save or restore operation are automatically cleared.

If tapes are used and a sequence number is specified, the tape is cleared and, starting with that sequence number, all tapes following the first tape are cleared.

***AFTER:** All the uncleared diskettes or tapes after the initial diskette or tape are automatically cleared. This option is not valid for save/restore operations to save files. If the operation cannot proceed because the first diskette or tape is uncleared, an inquiry message is sent to the system operator, allowing him to end the operation or to specify that the currently selected diskette or tape be cleared so the operation can continue. All diskettes or tapes used to perform the save operation should be cleared before the save command is issued.

PRECHK Parameter

Specifies if the save operation for a library should end if all objects in the library specified by this command do not satisfy the following conditions of the save operation: the objects exist, they were not previously found to be damaged, they are not locked by another job, and the requester of the save operation has authority to save the objects.

Note: If PRECHK(*YES) is specified, LIB(*NONSYS) cannot be specified. When multiple libraries are specified, PRECHK(*YES) applies to each library independently. For example, the failure to save one library does not affect the saving of other libraries.

***NO:** Allows the save operation for a library to continue, saving only those objects that can be saved.

***YES:** Ends the save operation for a library before any data is written if, after all objects in the specified library are checked, one or more objects cannot be saved. If multiple libraries are specified, the save operation continues with the next library.

ACCPTH Parameter

Specifies if the access paths dependent on the physical files being saved are also saved. The access paths are saved only if all members on which the access paths are built are included in this save operation. The access paths must be MAINT(*IMMED) or MAINT(*DLY), and the save operation must be able to get a *SHRNUP lock on the logical file containing the access path. If any of the these conditions are not met, the access path is not saved, but the save of the physical files proceeds normally. Information messages are sent indicating the number of logical file access paths saved with each physical file. The physical files on which the access path is built must exist in the same library. This parameter does not save logical file objects; it only controls the saving of the access paths. If logical files are also saved in the same command, the saving of the access paths is still determined by this parameter. More information on the restoring of saved access paths is in the *Data Base Guide*.

***NO:** Only those objects in the libraries specified on the command are saved. No logical file access paths are saved.

***YES:** The physical files in the specified libraries and all eligible access paths over them are saved.

STG Parameter

Specifies if the system storage that is occupied by the data portion of files, programs, and journal receivers in the library being saved is freed as part of the save operation. Only the data portion of the objects is freed, not the descriptions of the objects.

***KEEP:** The storage occupied by the data portion of the objects being saved is not freed.

***FREE:** The storage occupied by the data portion of the files (except for save files), programs, and journal receivers being saved is freed as part of the save operation. The storage for all the objects in a library is freed only after all the objects in that library are saved successfully.

DTACPR Parameter

Specifies if data compression is used.

***NO:** No data compression is performed.

***YES:** Program data compression is performed.

Examples

```
SAVLIB   LIB(JOE)  DEV(TAP1)
```

This command saves the library named JOE on whatever tape is put on tape device TAP1. The storage occupied by JOE in the system is not freed.

```
SAVLIB   LIB(QGPL)  DEV(DKT1) VOL(ABC DEF GHI)
```

The general purpose library (QGPL) is saved on the diskettes in the DKT1 device. The diskettes used must have the volume names ABC, DEF, and GHI. If the save operation is not finished when diskette ABC is full, a message is issued to the operator asking for volume DEF to be put on.

```
SAVLIB   LIB(CUSTDATA)  DEV(DKT1)  VOL(CUSNM CUSAD)  +
  STG(*FREE)
```

The library CUSTDATA is saved on volumes CUSNM and CUSAD, which are put in the DKT1 diskette device. The storage occupied by the files, programs and journal receivers in the CUSTDATA library is freed after it is saved.

```
SAVLIB   LIB(QGPL)  DEV(TAP1 TAP2 TAP3)  +
  VOL(USRA USRB USRC USRD) ENDOPT(*UNLOAD)
```

The library USRLIB is saved on four tape volumes on 3 tape devices. Volume USRA is put on device TAP1, volume USRB on TAP2, volume USRC on TAP3, and USRD on TAP1. Volume USRA is rewound, and must be unloaded by the operator when its processing is complete so TAP1 can be used for volume USRD.

```
SAVLIB  LIB(LIB1)  DEV(TAP1)  LABEL(MONDAYBACKUP)
```

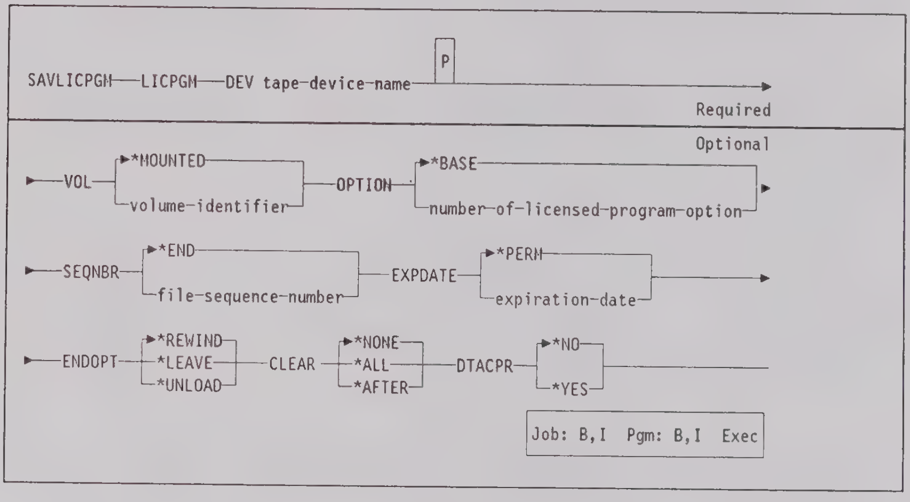
This command uses the tape device TAP1 to save the LIB1 library on tape. The library is saved with a media file label of MONDAYBACKUP. This label must be specified when restoring the library or any of its objects.

SAVLICPGM (Save Licensed Program) Command

The Save Licensed Program (SAVLICPGM) command saves a copy of all of the objects making up a licensed program. It saves the licensed program in a form that is restorable by the Restore Licensed Program (RSTLICPGM) command.

To save the system data on tape, the system writes a copy of all objects to save on tape.

Restrictions: (1) To use this command, you must have the save system authority in your user profile, specified by SPCAUT(*SAVSYS). (2) A standard labeled tape volume must be on the tape device. (3) Some licensed programs can only be saved if you are enrolled in the system distribution directory. See the publications for each licensed program for a description of this restriction.



LICPGM Parameter
Specifies which licensed program is saved. Select one of the following licensed program ID's:

5728AP1: The advanced double-byte character printer set licensed program is saved.

5728BA1: The BASIC licensed program is saved.

5728CB1: The COBOL licensed program is saved.

5728CM1: The communication licensed program is saved.

5728CR1: The cryptographic support licensed program is saved.

5728DB1: The S/38 utilities licensed program is saved.

5728DCT: The language dictionary licensed program is saved.

5728DS1: The graphic support licensed program is saved.

5728PC1: The personal computer (PC) licensed program is saved.

5728PL1: The PL/I licensed program is saved.

5728PS1: The PASCAL licensed program is saved.

5728PT1: The advanced performance tools licensed program is saved.

5728PW1: The programming development aids licensed program is saved.

5728QU1: The Query licensed program is saved.

5728RG1: The RPG licensed program is saved.

5728SS1: The optional part of the operating system in the OPTION parameter is saved.

5728ST1: The Structured query language program is saved.

5728WP1: The office support licensed program is saved.

DEV Parameter
Specifies the names of the tape devices used for the save licensed program operation. Each device name must already be known on the system by a device description. If multiple devices are specified, they must have compatible media formats. If more than one tape device is used, specify the names of the devices in the order in which they are to be used. A maximum of four device names can be specified. When more than one tape volume is used, you may want to use more than one tape device so that one tape volume can rewind unload while another tape device is processing.

VOL Parameter

Specifies the volume identifiers of the tape volumes on which the licensed program data is saved. The volumes must be put in the same order as the volume identifiers are specified in this parameter. A maximum of 50 entries can be specified. (More information on the VOL parameter is in Appendix A, "Expanded Parameter Descriptions.")

***MOUNTED:** The licensed program data is saved on whatever volumes are put on the device.

volume-identifier: Specify the identifiers of one or more volumes in the order they are put on the device and used to save the licensed program.

OPTION Parameter

Specifies which of the optional parts of the licensed program given in the LICPGM parameter is to be saved.

***BASE:** Specifies only the base part of the licensed program is to be saved.

number-of-licensed-program-option: Specify the number of the optional part of the listed licensed program that is to be saved.

SEQNBR Parameter

Specifies, only when tape is used, which sequence number is used for the restore process.

***END:** The system saves the file to the sequence number after the last sequence number on the tape.

file-sequence-number: Specify the sequence number of the file used for the save process.

EXPDATE Parameter

Specifies the end date of the tape save/restore files. If a date is specified, the files are protected and cannot be overwritten until the specified expiration date.

***PERM:** The save/restore files are protected permanently.

expiration-date: Specify the date when the save/restore files expire.

ENDOPT Parameter

Specifies the positioning operation to be done automatically on the tape volume when the save operation is completed. In the case of a multiple-volume save operation, this parameter applies to the *last* volume only; all other volumes are rewound and unloaded when the end of the tape is reached.

***REWIND:** The tape is rewound, but not unloaded, when the save operation is completed.

***LEAVE:** The tape should be left in its current position when the operation is completed; it is not rewound or unloaded.

***UNLOAD:** The tape is rewound and unloaded when the save operation is completed.

CLEAR Parameter

Specifies if uncleared tapes encountered during the save operation are automatically cleared. An uncleared tape is one containing a file with an expiration date later than the date of the save operation (which includes files protected permanently using EXPDATE(*PERM)). (This parameter does not control initializing the tapes; tapes should be standard labeled.) If an un-initialized tape volume is encountered during the save operation, an inquiry message allows the volume to be initialized.

***NONE:** None of the uncleared tapes encountered during the save operation are automatically cleared. If the save operation cannot proceed because an uncleared tape is encountered, an inquiry message is sent to the operator, allowing him to end the save operation, or to specify that the currently selected tape be cleared so the operation can continue. All tapes to be used to perform the save operation should be initialized before the save command is issued.

***ALL:** All uncleared tapes encountered during the save operation are automatically cleared.

If a sequence number is specified, the tape is cleared and, starting with that sequence number, all tapes following the first tape are cleared.

***AFTER:** All the uncleared tapes after the first tape are automatically cleared. If the operation cannot proceed because the first tape is uncleared, an inquiry message is sent to the system operator, allowing him to end the operation or to specify that the currently selected tape be cleared so the operation can continue. All tapes used to perform the save operation should be initialized before the save command is issued.

DTACPR Parameter

Specifies if data compression is used.

***NO:** No data compression is performed.

***YES:** Data compression is performed.

Examples

```
SAVLICPGM  LICPGM (5728WP1)  DEV(TAP1)  CLEAR(*ALL)
```

This command saves the Office licensed program. It is saved on the tape put on the TAP1 tape drive. Each uncleared tape is cleared automatically when it is encountered, and the save operation continues without operator intervention.

```
SAVLICPGM   LICPGM  (5728WP1)  DEV(TAP1)  VOL(ABCDE)
```

The Office licensed program is saved on the TAP1 tape drive, starting on the tape volume labeled ABCDE. If the save operation exceeds the storage capacity of one tape, a message requesting that another volume be put on the TAP1 tape drive is shown to the operator.

```
SAVLICPGM   LICPGM  (5728SS1)  OPTION(2)  DEV(TAP1)
```

The licensed program is saved on tape drives TAP1 and TAP2 in alternating order. If the save operation exceeds the storage capacity of two tapes, a message requesting that another volume be put on TAP1 is shown to the operator. The tapes are rewound at the completion of the save system operation.

The online help option of the OS/400 operating system is saved to the tape put on tape drive TAP1.

SAVOBJ (Save Object) Command

The Save Object (SAVOBJ) command saves a copy of a single object or a group of objects located in the same library. When OBJ(*ALL) is specified, objects can be saved from up to 50 libraries. When saving to a save file, only one library can be specified. The types of objects that can be saved by this command are listed in the OBJTYPE parameter. The system saves the specified objects by writing a copy of each one on diskettes, tapes, or in a save file. The objects are not affected in the system unless the command specifies that the storage should be freed. However, the description of each object is changed with the date, time, and place when it was last saved (unless UPDHST(*NO) is specified when saving to a save file).

For job queues, output queues, data queues, message queues, logical files, and save files, only the object descriptions are saved, and the contents of the objects are not saved. (However, logical file access paths can be saved by the use of the ACCPTH parameter.) The contents of a save file can be saved by use of the Save Save File Data (SAVSAVFDTA) command.

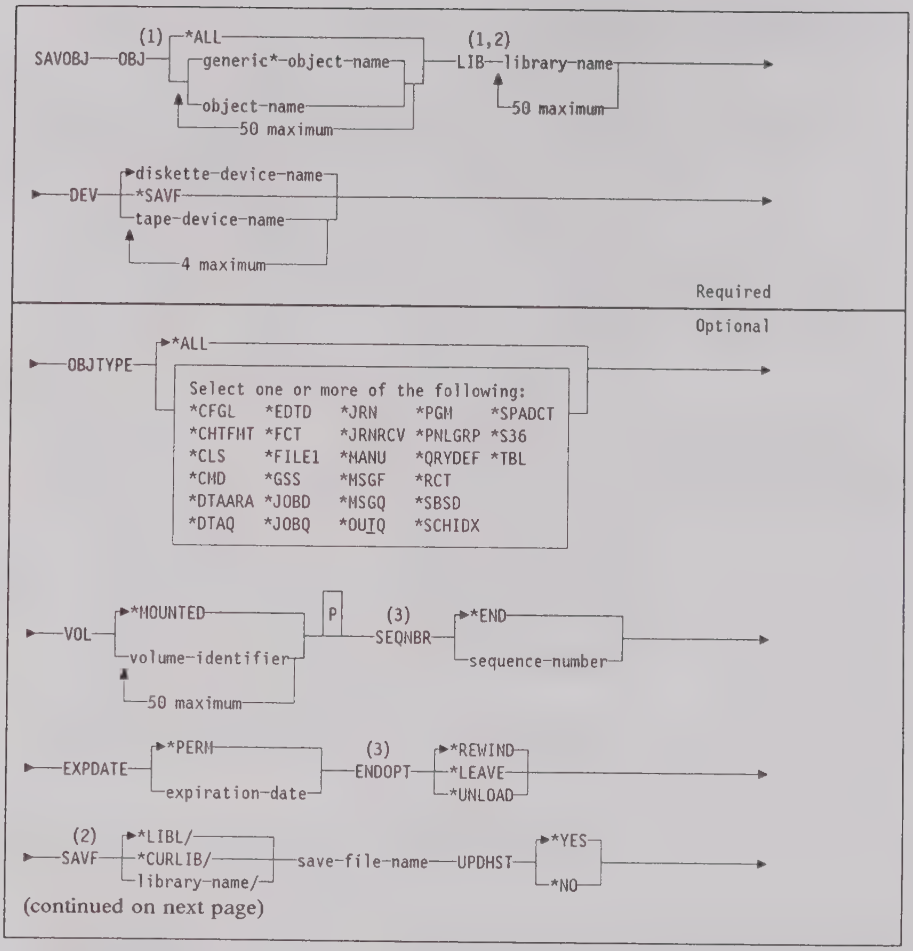
Note: This command ignores all file overrides that are currently in effect for the job.

Restrictions: To use this command, you must have either the special authority *SAVSYS specified in your user profile by the SPCAUT parameter, or you must have (1) object existence authority for each object specified, and (2) read authority for the specified library. (If you do not have the necessary authority to a specified object, all objects except that one are saved.) (3) SAVOBJ does not save the data dictionary for the library or its associated data base files. To save them, the SAVLIB command should be used.

When saving to tape or diskette, you must have use authority for the device description and device file. When saving to a save file, you must have add and use authority for the save file.

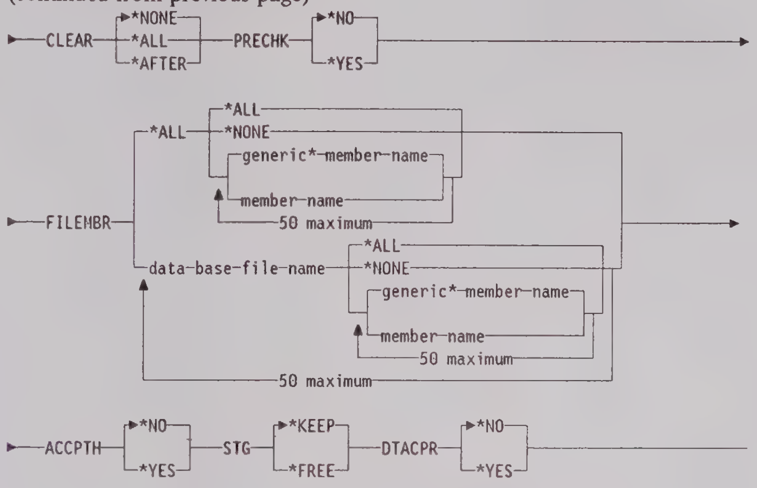
All diskettes to be used to save the objects must be initialized in the save/restore format. If tape is used, a standard labeled volume must be put on. Also, no object being saved can be changed by a job that is running at the time the save operation occurs.

If more than 8000 internal objects are saved from a specified library, the save operation ends for that library before any objects are saved, and an error message is sent to the system operator. This can occur if the library contains more than 8000 internal objects and OBJ(*ALL) is specified, or if the library contains more than 8000 objects whose generic names and object types are specified by the OBJ and OBJTYPE parameters.



(continued on next page)

(continued from previous page)



(1) OBJ(*ALL) is required when more than one library is specified.
(2) Only one library can be specified when SAVF is used.
(3) Applies to tape devices only.

Job: B,I Pgm: B,I Exec

OBJ Parameter

Specifies the names of one or more objects to save. All the objects must be in the library specified in the LIB parameter. Specify the name of each object or the generic name of each group of objects to be saved.

If the OBJTYPE parameter is not specified, all the object types listed in the description of the OBJTYPE parameter are saved, provided they are in the specified library and have the specified names.

***ALL:** All the objects in the specified libraries are saved, depending on the values specified for OBJTYPE.

generic-object-name:* Specify one or more generic names of groups of objects in the specified library to be saved.

object-name: Specify one or more names of specific objects to save. (Both generic names and specific names can be specified in the same command.)

LIB Parameter

Specifies which libraries contain the objects to be saved. Enter the name of the library. If *ALL is specified for the OBJ parameter, up to 50 library names can be entered.

Notes:

1. When multiple library names are specified, the overall save time is reduced, compared to saving objects from each of the libraries on individual commands. The best performance can be obtained by specifying the list of library names, in ascending order, by number of objects and/or members to be saved. For example, the first library saved should have the fewest number of objects and/or members to be saved.
2. Only one library can be specified when saving to a save file.

DEV Parameter

Specifies the name of the device on which the objects are saved. The device name must already be known on the system by a device description.

diskette-device-name: Specify the name of device used to save or restore the objects.

***SAVF:** The save or restore operation is done using the save file specified by the save file (SAVF) parameter.

tape-device-name: Specify the names of one or more tape devices used for the save or restore. If you are using more than one tape device (up to a maximum of four), enter the names of the devices in the order they are used. If you are using more than one tape volume, you may want to use more than one tape device so that one tape volume can rewind and/or unload while another tape device is processing.

OBJTYPE Parameter

Specifies the types of OS/400 objects to save.

***ALL:** All object types that are specified by name and are in the specified library are saved. If *ALL is also specified on the OBJ parameter, then all the objects in the library that are of the types shown in the following table are saved.

object-type: Specify the value for each of the types of objects to be saved. The following types can be specified:

Value	Type of CPF Object
*CFGL	Configuration list
*CHTFMT	Chart format
*CLS	Class
*CMD	Command
*DTAARA	Data area
*DTAQ	Data queue
*EDTD	Edit description
*FCT	Forms control table
*FILE[1]	File
*GSS	Graphics symbol set
*JOBD	Job description
*JOBQ	Job queue
*JRN	Journal
*JRNRCV	Journal receiver
*MENU	Menu definition
*MSGF	Message file
*MSGQ	Message queue (user-defined only)
*OUTQ	Output queue
*PGM	Program
*PNLGRP	Panel group
*QRYDEF	Query definition
*RCT	Reference code translation table
*SBSD	Subsystem description
*SCHIDX	Search information search index
*SPADCT	Spelling aid dictionary
*SSND	Session description
*S36	S/36 Machine description
*TBL	Table

[1] For logical files, and save files, only the definition of the file is saved. To save the access path for a logical file, use the ACCPTH parameter. To save the contents of a save file, use the Save Save File Data (SAVSAVFDTA) command.

[2] The description of the queue is saved, but not the information contained in the queue.

The object types shown are also the ones that are saved and restored by the Save Changed Object (SAVCHGOBJ), Save Library (SAVLIB), Restore Object (RSTOBJ), and the Restore Library (RSTLIB) commands. Data dictionaries and the associated files are only saved using SAVLIB.

VOL Parameter

Specifies the volume identifiers of the tape volumes or diskette volumes on which the object data is saved. The volumes must be on in the same order as the volume identifiers are specified in this parameter. A maximum of 50 entries can be specified. (More information on the VOL parameter is in Appendix A, "Expanded Parameter Descriptions.")

***MOUNTED:** The objects to be saved are copied on whatever volumes are on the device.

volume-identifier: Specify the identifiers of one or more volume in the order in which they are on the device and used to save the objects.

SEQNBR Parameter

Specifies, only when tape is used, which sequence number to use for saving the objects.

***END:** The system saves the objects after the last sequence number on the tape volume.

sequence-number: Specify the sequence number of the file to use as a starting point to save the objects.

EXPDATE Parameter

Specifies the expiration date of the save/restore files. If a date is specified, the save/restore files are protected and cannot be overwritten until the specified end date. This parameter is valid for both tape and diskette.

***PERM:** The save/restore files are protected permanently.

expiration-date: Specify the date on which the save/restore file(s) are no longer protected.

Note: For operations to diskette, the expiration date specified must be later than the date of the save operation. Otherwise, the save/restore files whose expiration date has been exceeded may be lost when the next save/restore file is written during the save operation.

ENDOPT Parameter

Specifies, only when tape is used, what operation is automatically done on the tape volume after the save operation ends. If more than one reel is included, this parameter applies only to the last reel used; all other reels are rewound and unloaded when the end of the tape is reached.

Note: If no objects are saved (for example, no objects meet the selection value specified on the command), the tape volume is not opened and the ENDOPT parameter is ignored.

***REWIND:** The tape is automatically rewound, but not unloaded, after the save operation has ended.

***LEAVE:** The tape is not rewound; another save operation can start at the current position on the tape.

***UNLOAD:** The tape is automatically rewound and unloaded after the save operation has ended.

SAVF Parameter

Specifies the qualified name of the save file used to contain the save data. The save file should be empty, unless CLEAR(*ALL) is specified.

The possible library values are:

***LIBL:** The library list is used to locate the save file.

***CURLIB:** The current library for the job is used to locate the save file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the save file is located.

UPDHST Parameter

Specifies if the save/restore history information of each object saved is changed with the date, time, and location of the save operation. The save history information for an object is available using the Display Object Description (DSPOBJD) command. The save history information is used to determine which journal entries are processed when RCVRNG(*LASTSAVE) and FROMENT(*LASTSAVE) are used on the Apply Journaled Changes (APYJRNCHG) command.

Note: This parameter only applies when saving to a save file (SAVF parameter).

***YES:** The last save date, time, and location information is changed for each object saved. This value works the same way as a save operation to diskette or tape.

***NO:** The save history information contained in the description of each object saved should not be changed.

Note: UPDHST(*NO) should only be used for a save operation that is not intended for recovery or save. For example, if the save data is sent, record by record, to another system and the save file immediately deleted, you probably do not want to update the save history information.

CLEAR Parameter

Specifies if an uncleared save file or uncleared diskettes or tapes encountered during the save operation are automatically cleared. An uncleared diskette or tape is one containing a file with an expiration date later than the date of the save operation (which includes files protected permanently using (EXPDATE(*PERM)). This parameter does not control initializing the diskettes or tapes; diskettes should already be initialized in the save/restore format; tapes should be standard labeled. If a diskette that is not initialized, is incorrectly named, or a tape volume that is not initialized is encountered during the save operation, an inquiry message allows the volume to be initialized or renamed.

***NONE:** None of the uncleared tapes, diskettes, or save files encountered during the save operation are automatically cleared. If the save operation cannot proceed because an uncleared diskette, tape, or save file is encountered, an inquiry message is sent to the operator, allowing him to end the save operation, or to specify that the currently selected diskette, tape, or save file be cleared so the operation can continue. If a save file is not cleared, the inquiry message is sent to the work station message queue for an interactive job, or to the operator for a batch job. The save file should be empty, or all diskettes or tapes to be used to perform the save operation should be cleared before the save command is issued.

***ALL:** All uncleared diskettes, tapes, or save files tapes encountered during the save operation are automatically cleared.

If tapes are used and a sequence number is specified, the tape is cleared and, starting with that sequence number, all tapes following the first tape are cleared.

***AFTER:** All the uncleared diskettes or tapes after the initial diskette or tape are automatically cleared. This option is not valid for save/restore operations to save files. If the operation cannot proceed because the first diskette or tape is uncleared, an inquiry message is sent to the system operator, allowing him to end the operation or to specify that the currently selected diskette or tape be cleared so the operation can continue. All diskettes or tapes used to perform the save operation should be cleared before the save command is issued.

PRECHK Parameter

Specifies if the save operation for a library should end if all objects specified by this command do not satisfy the following conditions: the objects exist, they were not previously found to be damaged, they are not locked by another job, and the requester of the save operation has authority to save the objects.

***NO:** Allows the save operation for a library to continue, saving only those objects that can be saved.

***YES:** Ends the save operation for a library before any data is written if, after all specified objects are checked, one or more objects cannot be saved. If multiple libraries are specified, the save operation continues with the next library.

FILEMBR Paramter

Specifies the data base file members to be saved. This parameter is made up of two parts: the file name and the member name. This parameter cannot be specified if MBROPT(*MATCH) is specified.

The file name values are:

***ALL:** The member list following this value applies to all files indicated by the OBJ parameter.

data-base-file-name: Specify the name of the data base file that contains the specified members to be saved. There may be up to 50 of the file/member list combinations specified for a single command.

Restrictions:

- Each data base file specified on the FILEMBR parameter must also be specified on the OBJ parameter by either its complete name, a generic name, or *ALL.
- The OBJTYPE parameter must be *ALL, or it must include *FILE.
- Generic names are not valid for the data base file name, but are allowed for the member name.
- Duplicate file names are not allowed.

The member name values are:

***ALL:** All members are saved from the specified file.

***NONE:** No members are saved from the specified file.

generic-member-name:* Specify the generic name of the members to be saved from the specified file.

member-name: Specify the specific name of the members to be saved from the specified file.

Restrictions:

- If the nongeneric member names are specified, all specified members must exist in the file in order for any part of the file to be saved or restored.
- If generic member names are specified, the file is not saved if it does not contain any members that match the generic names. (For example, if you specify PAY* as a generic member name, and the system is unable to find a member whose name starts with PAY, the file is not saved.) If no files specified by the FILEMBR parameter are saved because no members with the specified generic name can be found, a diagnostic message is sent, the save ends, and an escape message is sent specifying the number of files not saved. If at least one of the files processed for the FILEMBR parameter contains a member with the specified generic name, the diagnostic message is not sent, and the number of files not saved is in the final completion message.
- The members are first checked using the MBROPT parameter. If no match is found, then they are checked against the list of FILEMBR parameter.

ACCPTH Parameter

Specifies if the access paths dependent on the physical files being saved are also saved. The access paths are saved only if all members on which the access paths are built are included in this save operation. The access paths must be MAINT(*IMMED) or MAINT(*DLY), and the save operation must be able to get a *SHRNUP lock on the logical files using the access path. If any of these conditions are not met, the access path is not saved, but the save of the physical files proceeds normally. Informational messages are sent indicating the number of logical file access paths saved with each physical file. The physical files on which the access path is built must exist in the same library. This parameter does not save logical file objects; it only controls the saving of the access paths. If logical files are also saved in the same command, the saving of the access paths is still determined by this parameter. More information on the restoring of saved access paths is in the *Data Base Guide*.

***NO:** Only those objects specified on the command are saved. No logical file access paths are saved.

***YES:** The specified physical files and all eligible logical file access paths over them are saved.

STG Parameter

Specifies if the system storage that is occupied by the data portion of the file, program, and journal receiver objects being saved is to be freed after the save operation is finished. Only the data portion of the object is freed, not the object's description.

***KEEP:** The storage occupied by the data portion of the objects being saved is not freed.

***FREE:** The storage occupied by the data portion of the file, (except for save files), program, and journal receiver objects is freed as part of the save operation. The storage is freed only after all objects are saved successfully.

DTACPR Parameter

Specifies if data compression is used.

***NO:** No data compression is performed.

***YES:** Program data compression is performed.

Examples

```
SAVOBJ   OBJ(PETE)  LIB(LIBX)  DEV(DKT1)
```

This command saves the object named PETE located in LIBX. If, for example, LIBX contains both a program and a file named PETE, both objects are saved. The storage occupied by the object is not freed because the STG parameter default (*KEEP) was assumed.

```
SAVOBJ  OBJ(MSTRPAY PAY*)  LIB(QGPL)  DEV(DKT1)  STG(*FREE)
```

The object named MSTRPAY, and all the objects whose names start with the characters PAY located in the general purpose library (QGPL), are saved. The objects are copied on diskettes. As part of the save operation, the system storage that was occupied by the data portion of the saved file program and journal receiver objects is freed.

```
SAVOBJ   OBJ(FILEA)  OBJTYPE(*FILE)  LIB(LIBY)  DEV(DKT1)  +
  VOL(TOM)  CLEAR(*ALL)
```

The file named FILEA in library LIBY is saved on the diskette. The diskette must have the volume identifier TOM on it. If the diskette was not cleared, it is cleared automatically before FILEA is saved on it.

SAVSAVFDTA (Save Save File Data) Command

The Save Save File Data (SAVSAVFDTA) command saves the contents of a save file to either diskette or tape. This command saves the save/restore data in the save file to the device in a way that allows you to restore objects directly from diskette or tape.

The information written on tape or diskette by this command is similar to the data that was previously written to the save file by the save command that originally created the save file data. This includes the object descriptions, object contents, and program temporary fix (PTF) status information that existed when the original save was done.

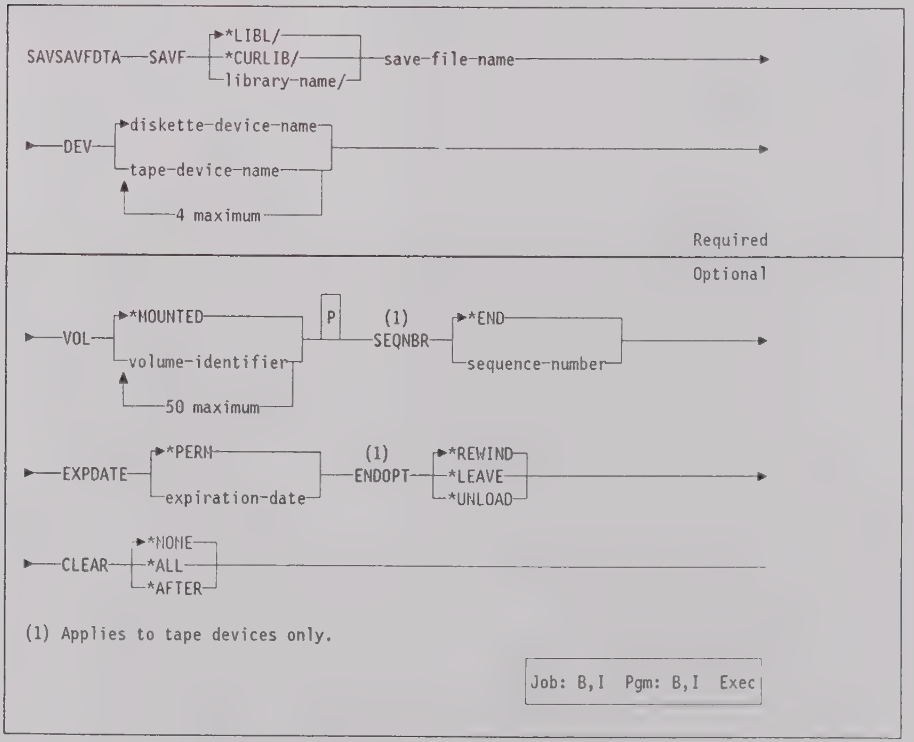
This command uses only the save file and device description objects; it does not refer to or modify the description or contents of the objects included in the file save data. Thus, objects included in the save file are not locked during the running of this command, and the save/restore history information (date, place, and time when each object was last saved) is not updated by this command for each object in the save file.

The description of the save file is not included in the save (unless it was included with the objects that were saved to create the save data in the file). In addition, this command does not update the save/restore history information for the save file object, so the last save date, time, and place always identify the last save of the save file object description, not its contents.

Note: This command ignores all file overrides currently in effect for the job.

Restrictions: To use this command, you must have use authority for the save file and use authority for the tape or diskette device description.

All diskettes to be used to perform the save operation must be initialized in the save/restore format. If a tape is used, a standard labeled tape volume must be put on. Also, the save file cannot be in use by a job running at the time the save operation occurs.



SAVF Parameter

Specifies the qualified name of the save file whose contents are saved. (If no library qualifier is specified, *LIBL is used to find the file.)

The possible library values are:

***LIBL:** The library list is used to locate the save file.

***CURLIB:** The current library for the job is used to locate the save file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the save file is located.

Note: The save file must not be empty; it must contain the data made by the run of a previous save command, or an error message is sent and its contents are not saved to diskette or tape.

DEV Parameter

Specifies the names of the devices used for the save operation. Each device name must be already known on the system by a device description.

diskette-device-name: Specify the name of the diskette device used to save the data.

tape-device-name: Specify the name of one or more tape devices that are used for the save operation. If more than one tape device is used, specify the names of the devices in the order they are to be used. A maximum of four device names can be specified. (When more than one tape volume is used, it may be advantageous to use more than one tape device so that rewind and/or unload of tape volumes can occur at the same time as processing on another tape device.)

VOL Parameter

Specifies the volume identifiers of the tape volumes (reels) or diskette volumes used for the save operation. The volumes must be in the same order as the volume identifiers are specified in this parameter. A maximum of 50 entries can be specified. (More information on the VOL parameter is in Appendix A, "Expanded Parameter Descriptions.")

***MOUNTED:** The save file data is copied on whatever volumes are on the device.

volume-identifier: Specify the identifiers of one or more volumes in the order they are put on the device and used to perform the save.

SEQNBR Parameter

Specifies, when tape is used, the sequence number on the tape where the save operation begins.

***END:** The data is saved to the next sequence number after the last sequence number that already exists on the tape.

sequence-number: The file data is saved to the sequence number specified on the command. The sequence number specified must either exist on the tape, or be one greater than the sequence number of the last file on the tape. Specifying a sequence number of 1 always causes the save operation to start at the beginning of the tape.

EXPDATE Parameter

Specifies the date that the file written on diskette or tape ends. The file is protected and cannot be written over until the specified end date.

PERM: The file written on diskette or tape is protected permanently.

expiration-date: The file on diskette or tape ends on the date specified on the command. The date must be specified in the format defined by the system values QDATFMT and, if separators are used, QDATSEP.

ENDOPT Parameter

Specifies, when tape is used, what operation is automatically performed on the tape volume after the save operation ends. If more than one reel is included, this parameter applies only to the last reel used. (This parameter is ignored if a diskette device is specified on the DEV parameter.)

***REWIND:** The tape is automatically rewound, but not unloaded, after the save operation ends.

***LEAVE:** The tape is not rewound; another save operation can start at the current position on the tape.

***UNLOAD:** The tape is automatically rewound and unloaded after the save operation ends.

CLEAR Parameter

Specifies if uncleared diskettes or tapes encountered during the save operation are automatically cleared. An uncleared diskette or tape is one containing a file with an expiration date later than the date of the save operation (which includes files protected permanently using EXPDATE(*PERM)). (This parameter does not control initializing the diskettes; they must already be initialized in the save/restore format, and tapes should be standard labeled.)

***NONE:** None of the uncleared tapes, diskettes, or save files encountered during the save operation are automatically cleared. If the save operation cannot proceed because an uncleared diskette, tape, or save file is encountered, an inquiry message is sent to the operator, allowing him to end the save operation, or to specify that the currently selected diskette, tape, or save file be cleared so the operation can continue. If a save file is not cleared, the inquiry message is sent to the work station message queue for an interactive job, or to the operator for a batch job. The save file should be empty, or all diskettes or tapes used to perform the save operation should be cleared before the save command is issued.

***ALL:** All uncleared diskettes, tapes, or save files tapes encountered during the save operation are automatically cleared.

If tapes are used and a sequence number is specified, the tape is cleared and, starting with that sequence number, all tapes following the first tape are cleared.

***AFTER:** All the uncleared diskettes or tapes after the initial diskette or tape are automatically cleared. This option is not valid for save/restore operations to save files. If the operation cannot proceed because the first diskette or tape is uncleared, an inquiry message is sent to the system operator, allowing him to end the operation or to specify that the currently selected diskette or tape be cleared so the operation can continue. All diskettes or tapes used to perform the save operation should be cleared before the save command is issued.

Examples

```
SAVSAVFDTA  SAVF(ONLINE)  DEV(TAP1)  +
  SEQNBR(1)  CLEAR(*ALL)
```

This command saves the contents of save file ONLINE to the first file on the tape volume on device TAP1. Files that have not ended on either the first tape volume or on subsequent volumes are overwritten without an inquiry message because CLEAR(*ALL) is specified.

If in the last save command to save library USRLIB the save file was:

```
SAVLIB  USRLIB  SAVF(ONLINE)
```

Then the following command must be specified to restore the library from tape:

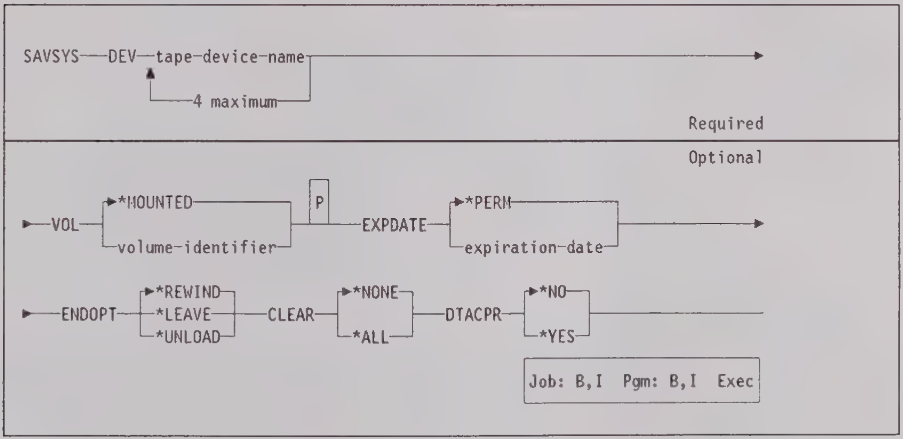
```
RSTLIB  USRLIB DEV(TAP1)  VOL(*MOUNTED)
```

SAVSYS (Save System) Command

The Save System (SAVSYS) command saves a copy of the programs and QSYS library in a format compatible with the installing of the AS/400 system. In addition, it saves the following objects: user profiles (including the object authorizations given to each profile); authorization lists, authority holders, and all the descriptions of devices, controllers, lines, modes, and classes of service.

To save the system data on offline storage, the system writes a copy of all objects to be saved on tape. The libraries and objects are not affected in the system; this command cannot be used to free any space occupied by these objects. The history information for the data area QSAVUSRPRF in QSYS is updated with the date, time, and place where user profiles were saved. The history information for the data area QSAVSYS in QSYS is updated with the date, time, and place where the system was saved. To display the information in these data areas, specify the Display Object Description (DSPOBJD) command, and specify DETAIL(*FULL). Save the information from the display of QSAVUSRPRF for the location where the user profiles were saved.

Restrictions: (1) To use this command, you must have the save system authorities in your user profile, specified by SPCAUT(*SAVSYS). (2) A standard labeled tape volume must be on the tape device. (3) The system cannot be running any other jobs during a save system operation (ENDSBS SBS(*ALL) must be issued).



DEV Parameter

Specifies the names of the tape devices used for the save system operation. Each device name must already be known on the system by a device description. If multiple devices are specified, they must have compatible media formats.

If more than one tape device is used, specify the names of the devices in the order in which they are used. A maximum of four device names can be specified. When more than one tape volume is used, you may want to use more than one tape device so that one tape volume can rewind and/or unload while another tape device is processing.

VOL Parameter

Specifies the volume identifiers of the tape volumes on which the system data is saved. The volumes must be put in the same order as the volume identifiers are specified in this parameter. A maximum of 50 entries can be specified. (More information on the VOL parameter is in Appendix A, "Expanded Parameter Descriptions.")

***MOUNTED:** The system data is saved on whatever volumes are put on the device.

volume-identifier: Specify the identifiers of one or more volumes in the order they are put on the device and used to save the system data.

EXPDATE Parameter

Specifies the end date of the tape save/restore files. If a date is specified, the files are protected and cannot be overwritten until the specified end date.

***PERM:** The save/restore files are protected permanently.

expiration-date: Specify the date when the save/restore files end.

ENDOPT Parameter

Specifies the positioning operation to be done automatically on the tape volume when the save operation is completed. In the case of a multiple-volume save operation, this parameter applies to the *last* volume only; all other volumes are rewound and unloaded when the end of the tape is reached.

***REWIND:** The tape is rewound, but not unloaded, when the save operation is completed.

***LEAVE:** The tape should be left in its current position when the operation is completed; it is not rewound or unloaded.

***UNLOAD:** The tape is rewound and unloaded when the save operation is completed.

CLEAR Parameter

Specifies if uncleared tapes encountered during the save operation are automatically cleared. An uncleared diskette or tape is one containing a file with an expiration date later than the date of the save operation (which includes files protected permanently using EXPDATE(*PERM)). (This parameter does not control initializing the tapes; tapes should be standard labeled.) If a tape volume that is not initialized is encountered during the save operation, an inquiry message allows the volume to be initialized.

***NONE:** None of the uncleared tapes encountered during the save operation are automatically cleared. If the save operation cannot proceed because an uncleared tape is encountered, an inquiry message is sent to the operator, allowing him to end the save operation, or to specify that the currently selected tape be cleared so the operation can continue. All tapes used to perform the save operation should be initialized before the save command is issued.

***ALL:** All uncleared tapes encountered during the save operation are automatically cleared. If a sequence number is specified, the tape is cleared and, starting with that sequence number, all tapes following the first tape are cleared.

DTACPR Parameter

Specifies if data compression is used.

***NO:** No data compression is performed.

***YES:** Data compression is performed.

Examples

```
SAVSYS  DEV(TAP1)  CLEAR(*ALL)
```

This command saves the system objects, all user profiles (including private authority for objects), and all line, control unit, and device descriptions. They are saved on the tape put on the TAP1 tape drive. Each uncleared tape is cleared automatically when it is encountered, and the save operation continues without operator intervention.

```
SAVSYS   DEV(TAP1)  VOL(ABCDE)
```

The system data is saved on the TAP1 tape drive, starting on the tape volume labeled ABCDE. If the save operation exceeds the storage capacity of one tape, a message requesting that another volume be put on the TAP1 tape drive is shown to the operator.

```
SAVSYS   DEV(TAP1 TAP2)
```

The system data is saved on tape drives TAP1 and TAP2 in alternating order. If the save operation exceeds the storage capacity of two tapes, a message requesting that another volume be put on TAP1 is shown to the operator. The tapes are rewound at the completion of the save system operation.

SAVS36F (Save System/36 File) Command

The Save System/36 File (SAVS36F) command creates a copy of a data base physical or logical file to diskette or tape magnetic media that can be restored on a System/36 or to a physical file that can be sent to a System/36. On a System/36, the file can be restored using the RESTORE system OCL procedure (or an equal operation of the $COPY SSP utility).

If saving to a data base physical file but the specified file (PHYFILE parameter) does not exist, it will be created.

This function is intended only for exchanging files with a System/36. For creating a backup version of a file, you should use AS/400 save commands (for example, Save Object (SAVOBJ) or Save Changed Object (SAVCHGOBJ)).

Restrictions:

- The following authorities are required (normally only applies when running on a system using resource security):
  - *USE authority for this command
  - *USE authority for the file specified in the FROMFILE parameter
  - *USE authority for the library specified in the FROMLIB parameter
  - *CHANGE authority to the file specified on the PHYFILE parameter if saving to an existing physical file.
  - *USE authority for the library specified in the PHYFILE parameter if saving to a physical file
  - *CHANGE authority for the library specified in the PHYFILE parameter if saving to a physical file and the file does not exist.
  - *USE authority for the diskette device description object, *USE authority for device file QSYSDKT, in library QSYS if saving to diskette
  - *USE authority for the tape device description object, *USE authority for device file QSYSTAP, in library QSYS if saving to tape
  - *USE authority for the based-on physical file if saving a logical file
- All diskettes that are used to save the file should be initialized using the INZDKT CL command or the equivalent System/36 Environment function (INIT OCL procedure or $INIT SSP utility). For a two-sided diskette, use a sector size of 256 or 1024. For a one-sided diskette, use a sector size of 128 or 512. If tape is used, each tape volume used should have been initialized with standard labels using the INZTAP CL command or the equivalent System/36 environment function (TAPEINIT OCL procedure or $TINIT SSP utility). Use a density of 1600 bits per inch when initializing the tape.

  Note: If the tape or diskette has not been initialized as stated above, the System/36 will not be able to process the media.

  If a tape or diskette is used that has not been properly initialized, a message is sent to the system operator that allows the operator to cancel the save or initialize the volume and continue.
- Object-level and record-level functions, other than read operations, should not be attempted for a file being saved by SAVS36F. Concurrent activity against the file (for example, moving the file or adding or removing records) can cause the save function to end if the file cannot be allocated.
- If saving a file to diskette, the diskette(s) used cannot contain an active file with the label name specified (TOLABEL parameter), because the AS/400 diskette functional manager does not allow duplicate diskette file labels.
- Not all physical and logical files can be saved with the SAVS36F command.
  - Only logical files created under the System/36 Environment (for example, through the BLDINDEX OCL procedure) or through a DDM request from a System/36 system can be saved. These files will be saved as System/36 alternative index files.
  - All physical files created under the System/36 Environment (for example, through the BLDFILE OCL procedure) or through a DDM request from a System/36

system will be saved using information stored within the AS/400 file description. These files are saved as System/36 sequential, direct, or indexed physical files.

  - Any physical files created by AS/400 commands or utilities can be saved as long as the record length is not greater than 4096. These files are saved as System/36 sequential files.

- To generate a save format which can be processed by the System/36 RESTORE procedure, the following information is not saved:
  - If saving a logical file, only the description of the file is saved. The index (or access path) is not saved.
  - If saving an indexed (keyed) physical file, the data is saved but the index is not. The index will be rebuilt after the file is restored.
- The specified file name (FROMFILE parameter) must meet AS/400 naming standards.

```
SAVS36F─────────────────────────────────────────────────────────────────►

                                               (1)  ┌─*PHYFILE────┐  ┌─┐
►──FROMFILE file-name──FROMLIB library-name──DEV─┤             ├──│P│──────►
                                                    └─device-name─┤  └─┘
                                                      ▲           │
                                                      └─4 maximum─┘
                                                                  Required
──────────────────────────────────────────────────────────────────────────
                                                                  Optional
                                    ┌►*LAST──────────────┐
►──TOLABEL file-label──CRTDATE──┤                    ├──────────────►
                                    └─file-creation-date─┘

              ┌►*END─────────────────┐       ┌►*MOUNTED──────────┐
►──SEQNBR──┤                      ├──VOL──┤                   ├──────►
              └─file-sequence-number─┘       └─volume-identifier─┤
                                               ▲                 │
                                               └───50-maximum────┘

              ┌►1────────────────┐         ┌►*REWIND─┐
►──RETAIN──┤                  ├──ENDOPT──┼─*LEAVE──┼─────────────────►
              └─retention-period─┘         └─*UNLOAD─┘

               ┌►*LIBL/─────────┐                      ┌►*NOREPLACE─┐
►──PHYFILE──┼─*CURLIB/───────┼──file-name──MBROPT──┤            ├──────
               └─library-name/──┘                      └─*REPLACE───┘

(1)  Up to 4 devices may be specified if more than one tape device is being
     used.  (Only one device is allowed with diskette.)

                                           ┌──────────────────────────┐
                                           │Job: B,I  Pgm: B,I  Exec  │
                                           └──────────────────────────┘
```

FROMFILE Parameter
Specifies the name of the file being saved. The file specified must exist in the library specified in the FROMLIB parameter.

FROMLIB Parameter
Specifies which library contains the data base file being saved.

DEV Parameter
Specifies the names of the devices used for the save operation. Each device name must already be known on the system by a device description.

*PHYFILE: Specifies the qualified name of the file that receives the copied file.

If a file by this name does not exist, it is created as a non-keyed, program-described file with a record length of 256. If a file already exists by this name, it is used as long as it is a non-keyed physical file with a record length of 256.

The copied records are put in the first member of the physical file. If the file has no members, a member is created using the name syntax 'Myymmdd' and the system date (for example, if the SAVS36F was run on June 28, 1987, the member name would be M870628).

device-name: Specify the name of the diskette device or the names of one or more tape devices (if installed) that are used for the save operation. If more than one tape device is used, enter the names of the devices in the order in which they are used. A maximum of four device names can be specified. When more than one tape volume is used, it may be advantageous to use more than one tape device so that rewinding and unloading of tape volumes can occur at the same time with processing on another tape device.

TOLABEL Parameter

Specifies the label given the new tape or diskette file created by the save operation. The TOLABEL name is also put into the System/36 file descriptor record which is stored at the beginning of the tape or diskette file. When the file is restored on a System/36, its name is the TOLABEL value (unless it is overridden by a //FILE OCL statement). A maximum of 8 characters can be used for the label.

If no value is specified, the value of the FROMFILE parameter is used as the diskette or tape label. If the FROMFILE parameter is an extended name, the characters between the quotes are used as the label. For example, specifying FROMFILE("MIKE&BOB") and no TOLABEL parameter would be the same as specifying TOLABEL('MIKE&BOB').

CRTDATE Parameter

For a *date-differentiated* file (maintained by the System/36 Environment), specifies which instance (member) of the file is saved. The default is to save the *LAST instance (most recently created member) of the specified file. (The date must be entered in the job date format (DATFMT); if separators, specified by the job date separator value (DATSEP), are used, the value must be enclosed in apostrophes.) This parameter is only valid if a data base physical file is being saved.

***LAST:** The most recently created member for the specified file is saved.

file-creation-date: Specify the creation date of the date-differentiated file member saved. The date specified is converted to year/month/day format and the member of the file (specified by the FILE parameter) with the name 'Myymmdd' is saved. For example, to save the instance of file 'FRED' created on July 10, 1986, specify FROMFILE(FRED) CRTDATE('07/10/86'), which would save member M860710 in file FRED.

This example allows a migrated System/36 user to identify the desired file/member very much like a System/36. However, it does not lend itself to letting the user save off a single member from an AS/400 multi-member file. For example, the user who has a file named FRED with members BOB and JOE could not pick which member should be saved (the user would have to copy the member to a new file and save the new file specifying CRTDATE(*LAST) or rename the member to follow the 'Myymmdd' name syntax).

SEQNBR Parameter

Specifies, only when tape is used, which sequence number is used for the save process.

***END:** The specified file is saved to the sequence number after the last sequence number on the tape (for example, this file goes after all other files that are on the volume).

file-sequence-number: Specify the sequence number of the tape file that is used when saving the specified file. If this sequence number already exists on the tape volume, the tape label at that sequence number must match the TOLABEL parameter. The existing file at that sequence number is overwritten, and all subsequent files on the volume are not accessible after the save.

If a new tape file is added to the tape, the sequence number must be one higher than the sequence number of the last tape file on that volume. No gaps are allowed in the series of sequence numbers.

VOL Parameter

Specifies the volume identifiers of the tapes or diskettes used for saving the file. If a diskette or tape volume label does not match the specified volume identifier, an inquiry message is sent to the system operator. The message options are Cancel (end the save), Ignore (go ahead and use the diskette or tape), Retry (make the volume check again on the diskette or tape that has just been put on the device), or Initialize (initialize the volume to the expected volume name).

***MOUNTED:** The file to be saved is copied on whatever volume is on the device. For diskettes, use whatever volume is loaded in the diskette drive. If the diskette loaded in the diskette drive is full of active files, the save ends.

volume-identifier: Specify the volume identifiers of the tapes or diskettes used for saving the file. For each tape or diskette volume name, a maximum of 6 characters can be specified using any combination of letters and numbers. The user should ensure that the volume name meets the System/36 volume identifier requirements. A maximum of 50 volume identifiers can be specified.

RETAIN Parameter

Specifies the retention period for the newly created tape or diskette file. The file is protected and cannot be written over until the day after the retention period ends.

1: The default retention period is one day.

retention-period: Specify the number of days the tape or diskette file should be kept. Allowable values are 0 through 999. If a retention period of 999 is specified, the tape or diskette file becomes a permanent file.

ENDOPT Parameter

Specifies, only when tape is used, what operation is automatically performed on the tape volume after the save operation ends. If more than one reel is used, this parameter applies only to the last reel used. (This parameter is ignored if a diskette device is specified in the DEV parameter.)

***REWIND:** The tape is automatically rewound, but not unloaded, after the save operation has ended.

***LEAVE:** The tape is not rewound; another tape operation can start at the current position on the tape.

***UNLOAD:** The tape is automatically rewound and unloaded after the save operation has ended.

PHYFILE Parameter

Specifies the qualified name of the file that receives the copied source file member data.

If a file by this name does not exist, it is created as a non-keyed, program-described physical file with a record length of 256. If a file already exists by this name, it is used as long as it is a non-keyed physical file with a record length of 256.

The copied records are put in the first member of the physical file. If the file has no members, a member is created using the name syntax 'Myymmdd' and the system date (for example, if the SAVS36LIBM was run on June 28, 1987, the member name would be M870628).

The possible library values are:

***LIBL:** The library list is used to locate the physical file.

***CURLIB:** The current library for the job is used to locate the physical file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the physical file is located.

MBROPT Parameter

Specifies if the data of an existing physical file member is replaced.

***NOREPLACE:** If a file already exists by the name specified on the PHYFILE parameter in the specified library, an error message is sent and the data in that member is not replaced.

***REPLACE:** The PHYFILE member is cleared before copying the first record.

Examples

```
SAVS36F  FROMFILE(PETE)  FROMLIB(QS36F)  DEV(I1)
```

This command saves the file named PETE located in library QS36F. Assuming that I1 is the name of a diskette device description, the file is saved on the diskette placed in the diskette drive. The diskette file label is PETE (same as the FROMFILE name). If PETE is a date-differentiated physical file, the most recently created instance (member) of PETE is saved. The diskette file has a retention period of one day (the retention period ends at midnight of the following day).

```
SAVS36F  FROMFILE(MSTRPAY)  FROMLIB(PAYLIB)  DEV(T1 T2)  +
  TOLABEL('PAY.MSTR')  RETAIN(999)
```

The file named MSTRPAY located in library PAYLIB is saved. Assuming that T1 and T2 are tape devices, the file is copied to the tapes on devices T1 and T2. The tape file label is PAY.MSTR and the tape file is a permanent file. The last tape used for the save is rewound at the end of the save operation.

SAVS36LIBM (Save System/36 Library Members) Command

The Save System/36 Library Members (SAVS36LIBM) command creates a copy of source file members in a file on diskette or tape that can be restored on a System/36, or into a data base physical file on AS/400 that can be sent to a System/36.

The saved member file is formatted like a record-mode LIBRFILE created on a System/36 using the System/36 FROMLIBR system OCL procedure (or the equivalent OCL call of the $MAINT SSP utility). On a System/36, the diskette or tape or disk file can be restored using the TOLIBR system OCL procedure (or the equivalent OCL call of the $MAINT SSP utility).

If saving to a data base physical file but the specified file (PHYFILE parameter) does not exist, it will be created.

This command is intended only for exchanging source and procedure data with a System/36. It provides a simplified command interface for an AS/400 customer who migrated from a System/36, but is not well-suited for backing up of AS/400 source files. You should use the AS/400 CL commands (SAVOBJ or SAVCHGOBJ) for creating a backup copy of an AS/400 source file or individual source file members.

Restrictions:

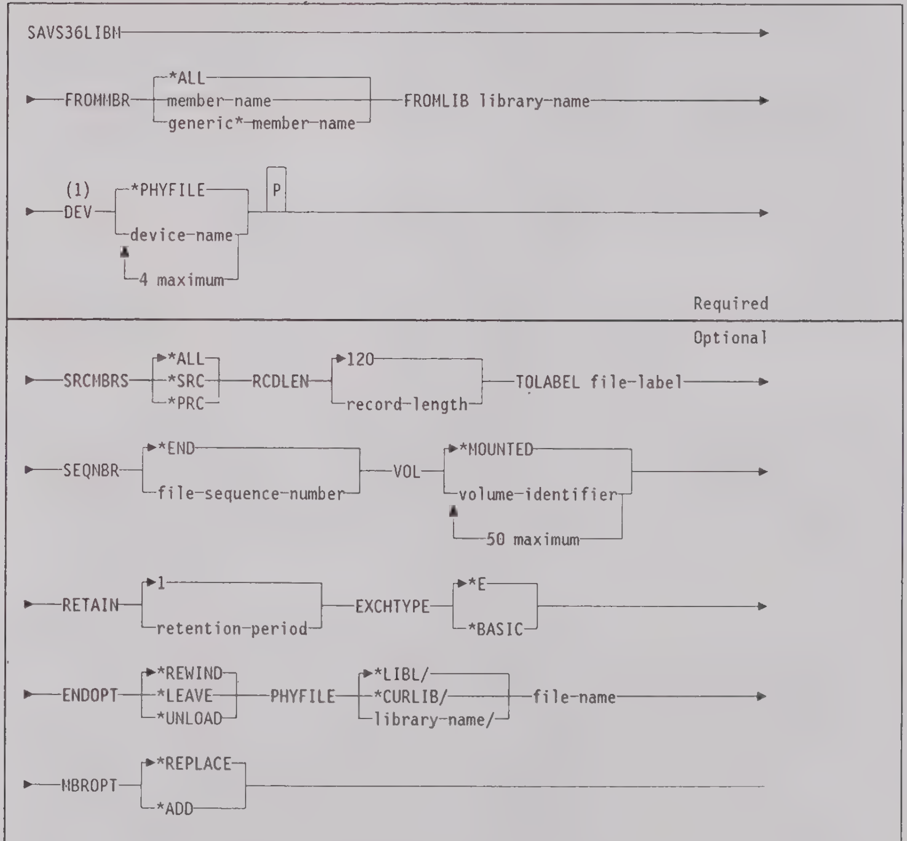
- The following authorities are required (normally only applies when running on a system using resource security):
  - *USE authority for this command
  - *USE authority for the library specified in the FROMLIB parameter
  - *USE authority for file QS36SRC in the specified library if saving source library members
  - *USE authority for file QS36PRC in the specified library if saving procedure library members
  - *USE authority for the library specified in the PHYFILE parameter if saving to a physical file
  - *CHANGE authority for the library specified in the PHYFILE parameter if saving to a physical file and the file does not exist
  - *CHANGE and *OBJMGMT authority for that file if saving to a physical file with MBROPT(*ADD)
  - *ALL authority for the file if saving to a physical file with MBROPT(*REPLACE)
  - *USE authority for the diskette device description object, *USE authority for device file QSYSDKT in library QSYS if saving to diskette
  - *USE authority for the tape device description object, *USE authority for device file QSYSTAP in library QSYS if saving to tape
- All diskettes that are used to save the members should be initialized using the INZDKT CL command or the equivalent System/36 environment function (INIT OCL procedure or $INIT SSP utility). For a two-sided diskette, use a sector size of 256 or 1024. For a one-sided diskette, use a sector size of 128 or 512. If tape is used, each tape volume used should first be initialized with standard labels using the INZTAP CL command or the equivalent System/36 environment function (TAPEINIT OCL procedure or $TINIT SSP utility). Use a density of 1600 bits per inch when initializing the tape.

  Note: If the tape or diskette has not been initialized as stated above, the System/36 will not be able to process the media.

  If a tape or diskette is used that has not been properly initialized, a message is sent to the system operator that allows the operator to cancel the save or initialize the volume and continue.
- Object-level functions, other than read operations, should not be used for files QS36SRC and QS36PRC while members are being saved by SAVS36LIBM. If the necessary file(s) cannot be allocated, no members are saved.

Record-level functions, other than read operations, should not be used for members being saved. Concurrent activity against a member (for example, adding or removing records) can cause the member to be omitted from the save.

- If saving a file to diskette, the diskette(s) used cannot contain an active file with the same name as the TOLABEL parameter value, because the AS/400 system does not allow duplicate diskette file labels.
- Only members from source files QS36SRC (for *SRC members) and QS36PRC (for *PRC members) in the specified library can be saved using the SAVS36LIBM command. Only the member data is saved from the source file member (that is, the source sequence number and change date fields are not saved).
- The specified member name or generic member name (FROMMBR parameter) must meet AS/400 naming standards. When saving a member that has an extended name, the quote characters are not stored as part of the member name in the output file (for example, member "A + B" would be saved as A + B).



(1) Up to four devices may be specified if more than one tape device is being used. (Only one device is allowed with diskette.)

Job: B,I Pgm: B,I Exec

FROMMBR Parameter

Specifies the names of the members being saved. Specify a single member name or a single generic member name or *ALL for the members to be saved.

***ALL:** All members are saved of the member type specified on the SRCMBRS parameter.

member-name: The members having the specified member name are saved.

generic-member-name:* All members which have the specified generic member name are saved.

FROMLIB Parameter
Specifies which library contains the members being saved.

DEV Parameter
Specifies the names of the devices used for the save operation. Each device name must already be known on the system by a device description.

***PHYFILE:** The output file is the data base physical file specified by the PHYFILE parameter.

device-name: Specify the name of the diskette device or the names of one or more tape devices that are used for the save operation. If more than one tape device is used, enter the names of the devices in the order in which they are used. A maximum of four device names can be specified. When more than one tape volume is used, it may be advantageous to use more than one tape device so that rewinding and unloading of tape volumes can occur at the same time with processing on another tape device.

SRCMBRS Parameter
Specifies which member types (source and procedure members on System/36) are saved.

***ALL:** Save System/36 source and procedure members (from QS36SRC and QS36PRC) which match the member name specified (FROMMBR parameter). For example, if you specify FROMMBR(PAY*) and SRCMBRS(*ALL), all members from both source files that begin with the letters 'PAY' will be saved.

***SRC:** Save only System/36 source members (from file QS36SRC) which match the member name specified (FROMMBR parameter).

***PRC:** Save only System/36 OCL procedure members (from file QS36PRC) which match the member name specified (FROMMBR parameter).

RCDLEN Parameter
Specifies the file record length, in bytes, used when copying the members. This parameter is ignored when saving members to an existing data base physical file.

120: This is the maximum record length allowed for System/36 source and procedure library members. Using this length ensures that no data is truncated when copying the records to the output file.

record-length: Specify a record length in bytes. The record length can be from 40 to 120. Specifying a record length less than 120 may cause data to be lost (truncated) for records longer than the specified record length.

TOLABEL Parameter
Specifies the label of the diskette or tape file that receives the copied source file member data. A maximum of 8 characters can be used for the label value for the diskette or tape file. If the DEV parameter is not *PHYFILE, a TOLABEL value must be specified.

SEQNBR Parameter
Specifies, only when tape is used, which sequence number is used for the save process.

***END:** The specified members are saved to the sequence number after the last sequence number on the tape (that is, this file goes after all other files currently on the tape volume).

file-sequence-number: Specify the sequence number of the tape file that is used when saving the specified members. If this sequence number already exists on the tape volume, the tape label at that sequence number must match the TOLABEL parameter. The existing data file at that sequence number is overwritten, and all subsequent files on the volume are not accessible after the save.

If a new tape file is added to the tape, the sequence number must be one higher than the sequence number of the last tape file on that volume. No gaps are allowed in the series of sequence numbers.

VOL Parameter
Specifies the volume identifiers of the tapes or diskettes used for saving the members. If a diskette or tape volume label does not match the specified volume identifier, an inquiry message is sent to the system operator. The message options are Cancel (end the save), Ignore (go ahead and use the diskette or tape which is on the device), Retry (make the volume check again on a new diskette or tape), or Initialize (initializes the volume to the expected volume name).

***MOUNTED:** The members saved are copied on whatever volume is on the device. For diskettes, use whatever volume is loaded in the diskette drive. If the diskette loaded in the diskette drive is full of active files, the save ends.

volume-identifier: Specify the volume identifiers of the tapes or diskettes used for saving the members. For each tape or diskette volume name, a maximum of 6 characters can be specified using any combination of letters and numbers. The user should ensure that the volume name meets the System/36 volume identifier requirements. A maximum of 50 volume identifiers can be specified.

RETAIN Parameter
Specifies the retention period for the newly created tape or diskette file. The file is protected and cannot be written over until the day after the retention period ends.

1: The default retention period is one day.

retention-period: Specify the number of days the tape or diskette file should be kept. Allowable values are 0 through 999. If a retention period of 999 is specified, the tape or diskette file becomes a permanent file.

EXCHTYPE Parameter

Specifies the exchange type for the newly created diskette file.

E: The default is to create an E-exchange diskette file. An E-exchange file is a system save file that can be restored either on a System/36 (using the TOLIBR procedure) or on an AS/400 (using the RSTS36LIBM command).

BASIC: The output diskette file is to be in basic exchange format. A basic exchange format file can be restored or copied to a System/34 or System/32 or any other system which supports the basic exchange diskette format.

ENDOPT Parameter

Specifies, only when tape is used, what operation is automatically performed on the tape volume after the save operation ends. If more than one reel is used, this parameter applies only to the last reel used. (This parameter is ignored if a diskette device is specified in the DEV parameter.)

REWIND: The tape is automatically rewound, but not unloaded, after the save operation has ended.

LEAVE: The tape is not rewound; another tape operation can begin at the current position on the tape.

UNLOAD: The tape is automatically rewound and unloaded after the save operation has ended.

PHYFILE Parameter

Specifies the qualified name of the file that receives the copied source file member data.

If a file by this name does not exist, it is created in the current library if a library name was not specified, as a non-keyed, program-described physical file with the specified record length (RCDLEN parameter). If a file already exists by this name, it is used as long as it is a non-keyed physical file with a record length from 40 to 120.

The copied records are put in the first member of the physical file. If the file has no members, a member is created using the name syntax 'Myymmdd' and the system date (for example, if the SAVS36LIBM was run on June 28, 1987, the member name would be M870628).

The possible library values are:

LIBL: The library list is used to locate the physical file.

CURLIB: The current library for the job is used to locate the physical file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the physical file is located.

file-name: Specify the name of the file that receives the copied source file member data.

MBROPT Parameter

Specifies if copied records replace existing records in the PHYFILE member, or if the copied records are added after all records in the PHYFILE member.

REPLACE: The PHYFILE member is cleared before copying the first record.

ADD: The copied records are to be added to the records existing in the receiving PHYFILE member.

Examples

```
SAVS36LIBM  FROMMBR(XYZ1)  FROMLIB(JOHNSON)  DEV(I1)  +
  SRCMBRS(*PRC)  TOLABEL(XYZ1)
```

The single OCL procedure member XYZ1 (in source file QS36PRC) in library JOHNSON is saved. Assuming I1 is a diskette device, the member is saved to diskette file XYZ1. The file length is 120 and the retention period is one day.

```
SAVS36LIBM  FROMMBR(X*)  FROMLIB(ORDER)  DEV(*PHYFILE)  +
  PHYFILE(NETLIB/S36SRC) MBROPT(*ADD)
```

All source and procedure members (members in QS36SRC and QS36PRC) with names starting with the character 'X' in library ORDER are saved into data base physical file S36SRC in library NETLIB. The copied records are added after any records already in the file.

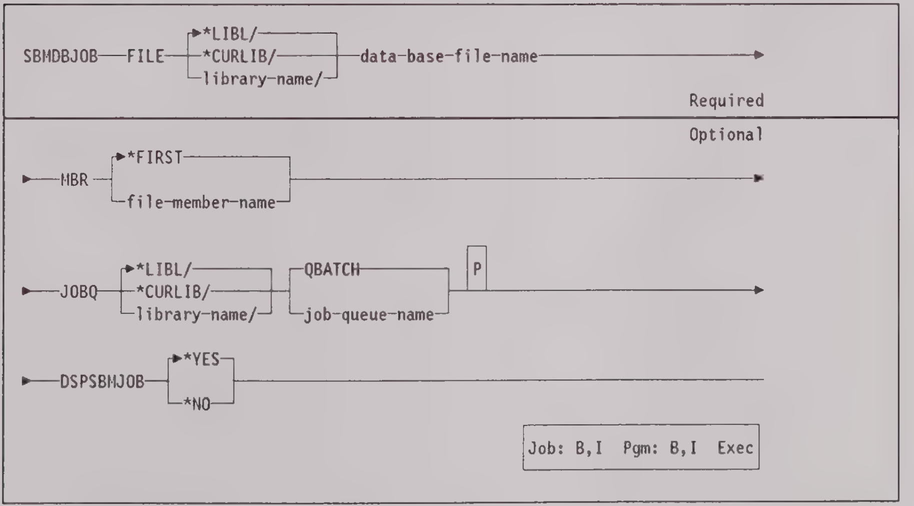
SBMDBJOB (Submit Data Base Jobs) Command

The Submit Data Base Jobs (SBMDBJOB) command allows a job that is running to submit other jobs to job queues to be run as batch jobs. The input stream is read either from a physical data base file or from a logical data base file in single-record format. This command specifies the name of the data base file and member that the jobs are read from, the name of the job queue to be used if the BCHJOB command has JOBQ(*RDR) specified, and whether jobs being submitted can be displayed by the Work with Submitted Jobs (WRKSBMJOB) command.

A Submit Data Base Jobs operation reads the file once and ends when the end-of-file is read or when an ENDINP (End Input) command (delimiter) is encountered. The ENDINP command (delimiter) is not recognized if it is within an inline file that ends with characters that are not default ending characters (as specified in the ENDCHAR parameter of the DATA command). The SBMDBJOB operation can be canceled either by canceling the request from the system request menu or by canceling the job that the process is running in.

In contrast to a spool reader started with the Start Data Base Reader (STRDBRDR) command, the SBMDBJOB command operates in the same process as the requesting function and does not do syntax checking on the input stream.

Restriction: The specified data base file must consist of single-field records and must have an arrival sequence access path, or it must be a standard data base source file.



FILE Parameter
Specifies the name of the data base file that the input stream is read from.

The possible library values are:

***LIBL:** The library list is used to locate the file.

***CURLIB:** The current library for the job is used to locate the file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the file is located.

MBR Parameter
Specifies the name of the member in the specified file that contains the input stream being read.

***FIRST:** No member name is specified; the first member in the file is used.

file-member-name: Specify the name of the member that contains the input stream being read.

JOBQ Parameter
Specifies the qualified name of the job queue that the jobs are placed on. An entry is placed on this queue for each job in the job stream that specifies JOBQ(*RDR) on its BCHJOB command. If *RDR is not specified on the BCHJOB command, the job queue specified in the BCHJOB command or in the job description is used. (Note that the job queue for each job in the input stream can be different.)

Restriction: If the user identified in the job description of the job being read is not authorized to the job queue that the job is placed on, that job ends, and a diagnostic message that describes the cause for ending is placed in the job log. The input stream, however, continues to be processed, starting with the next job.

***LIBL/QBATCH:** The job entry is placed on the QBATCH job queue, which is the default job queue, if JOBQ(*RDR) is specified on the BCHJOB command.

The possible library values are:

***LIBL:** The library list is used to locate the job queue.

***CURLIB:** The current library for the job is used to locate the job queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the job queue is located.

job-queue-name: The qualified name of the job queue that each job in the job stream is sent to if JOBQ(*RDR) is specified on the BCHJOB command.

DSPSBMJOB Parameter

Specifies whether the jobs being submitted are displayed on the submitted jobs display. Any submitted job of the type specified by the SBMFROM parameter of the WRKSBMJOB command can be displayed if *YES is specified on this parameter.

***YES:** This job can be displayed by the WRKSBMJOB command.

***NO:** This job is not displayed on any display produced by the WRKSBMJOB command.

Example

```
SBMDBJOB  FILE(QGPL/BILLING)
```

This command submits jobs using input from the data base file named BILLING, which is in the QGPL library. The first member in the BILLING file contains the input stream to be processed. The default job queue QBATCH is used.

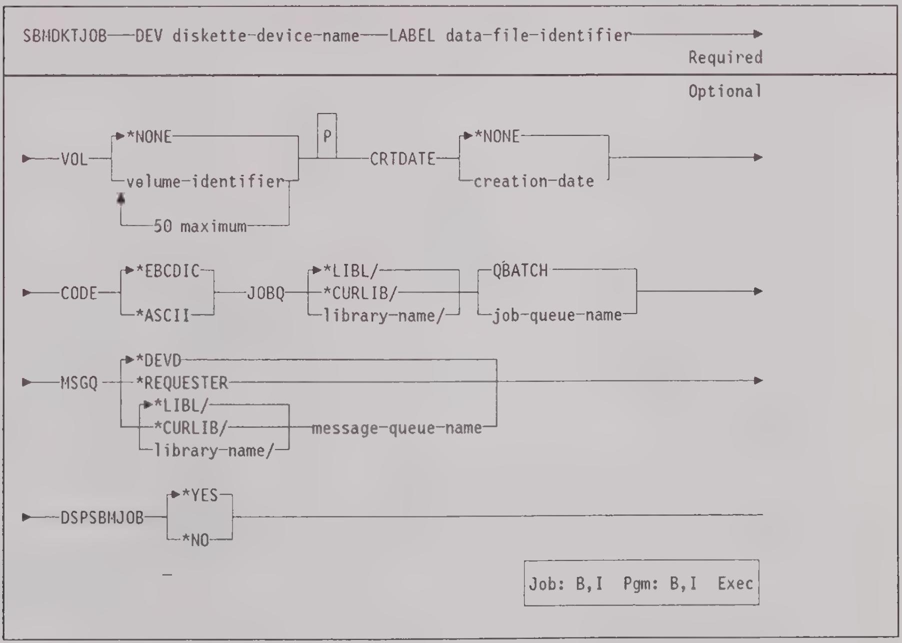
SBMDKTJOB (Submit Diskette Jobs) Command

The Submit Diskette Jobs (SBMDKTJOB) command allows jobs that are running to submit other jobs to job queues to be run as batch jobs. The input stream is read from a diskette. This command specifies the name of the diskette device, the label identifier, volume, creation date, and interchange code type of the file containing the input stream, the names of the job queue and message queue used, and whether jobs being submitted can be displayed by the Work with Submitted Jobs (WRKSBMJOB) command.

A Submit Diskette Jobs operation reads the file once and ends when the end-of-file is read or when an ENDINP (End Input) command (delimiter) is encountered. The ENDINP command (delimiter) is not recognized if it is within an inline file that ends with characters that are not default ending characters (as specified in the ENDCHAR parameter of the DATA command). The SBMDKTJOB operation can be canceled by either canceling the request from the system request menu or by canceling the job that the operation is running in.

In contrast to a spool reader started with the Start Diskette Reader (STRDKTRDR) command, the SBMDKTJOB command operates in the same process as the requesting function and does not do syntax checking on the input stream.

Note: This command cannot be used to read data files of diskettes that are in the save/restore format.



DEV Parameter
Specifies the name of the diskette device used to read the input stream.

LABEL Parameter
Specifies the data file label of the diskette data file on diskette that contains the input stream. The data file identifier can be no longer than 8 characters.

VOL Parameter
The volume (VOL) parameter specifies the volume identifiers of the volumes used in a diskette operation. A volume may consist of a single diskette or multiple diskettes. Normally, a volume is a single diskette.

***NONE:** No volume identifiers are specified; the current volume is used.

volume-identifier: The identifiers of one or more volumes are in the order that they are put on the device and used. Each volume identifier contains a maximum of 6 characters. A blank is used as the separator character when listing multiple identifiers.

CRTDATE Parameter

Specifies when the diskette data file was created on diskette; the creation date should not be specified if the date is not checked. If the date written on the diskette containing the data file does not match the date specified here, an error message is sent to the message queue named in the MSGQ parameter.

***NONE:** The creation date is not specified; no check is made.

creation-date: Specify the creation date of the data file to read. The date should be in the format specified by the QDATFMT system value.

CODE Parameter

Specifies the type of character code that is used by the diskette device.

***EBCDIC:** The EBCDIC character code is used.

ASCII: The ASCII character code is used.

JOBQ Parameter

Specifies the qualified name of the job queue that the job entries are to be placed on. An entry is placed on this job queue for each job in the job stream that specifies JOBQ(*RDR) on its BCHJOB command. If *RDR is not specified on the BCHJOB command, the job queue specified in the BCHJOB command or in the job description is used. (Note that the job queue for each job in the input stream can be different.)

Restriction: If the user identified in the job description of the job being read is not authorized to the job queue that the job should be placed on, that job ends, and a diagnostic message that describes the reason for ending is placed in the job log. The input stream, however, continues to be processed, starting with the next job.

The possible library values are:

***LIBL:** The library list is used to locate the job queue.

***CURLIB:** The current library for the job is used to locate the job queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the job queue is located.

QBATCH: The job entry is placed on the QBATCH job queue, which is the default job queue if JOBQ(*RDR) is specified on the BCHJOB command.

job-queue-name: Specify the qualified name of the job queue that each job in the job stream is sent to if JOBQ(*RDR) is specified on its BCHJOB command.

MSGQ Parameter

Specifies the qualified name of the message queue that operational messages are to be sent to.

***DEVD:** The messages are to be sent to the message queue indicated by the device description of the device being used.

***REQUESTER:** The messages are sent to the message queue of the user who started the process. This value is not valid for batch jobs.

The possible library values are:

***LIBL:** The library list is used to locate the message queue.

***CURLIB:** The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message queue is located.

message-queue-name: Specify the qualified name of the message queue that operational messages are being sent to.

DSPSBMJOB Parameter

Specifies whether the jobs being submitted are to be displayed on the submitted jobs display. Any submitted job of the type specified by the SBMFROM parameter of the WRKSBMJOB command can be displayed if *YES is specified.

***YES:** This job can be displayed by the WRKSBMJOB command.

***NO:** This job is not displayed on any display produced by the WRKSBMJOB command.

Example

```
SBMDKTJOB  DEV(QDKT)  LABEL(OCT24)  VOL(SALES)
```

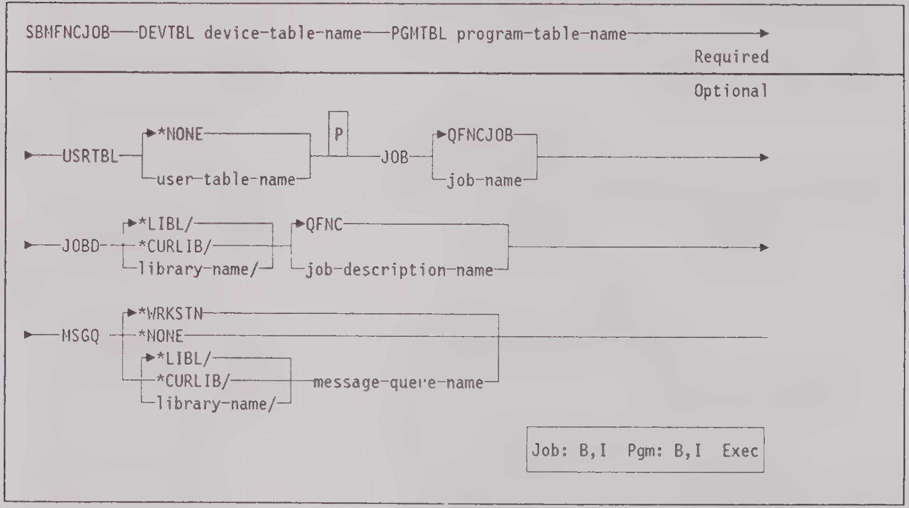
This command submits diskette jobs using diskette input from the device QDKT. The submit diskette jobs function gets its input from the data file named OCT24 with the volume identifiers SALES. The default job queue QBATCH is used as the receiving job queue when JOBQ(*RDR) is found in the job description. Operational messages are sent to the message queue defined by the device.

SBMFNCJOB (Submit Finance Job) Command

The Submit Finance Job (SBMFNCJOB) command submits a batch job that lets your finance application programs communicate with your 4701 control unit application programs.

You can use the SBMFNCJOB command only if:

- You are communicating with a 4701 control unit.
- You have defined a device table and a program table using the Work with Device Table (WRKDEVTBL) and Work with Program Table (WRKPGMTBL) commands. Defining a user table using the Work with User Table (WRKUSRTBL) command is optional.
- Your 4701 control unit application program sends data (transactions) first and expects a response.
- Your 4701 control unit application program passes data in the proper format.



DEVTBL Parameter

Specifies the name of the device table that the finance job uses to determine which 4700 devices it is to control.

PGMTBL Parameter

Specifies the name of the program table that the finance job uses to determine, from the program ID (sent in the data stream with a 4700 transaction), which system user program names will process the 4700 transaction.

USRTBL Parameter

Specifies the name of the user table that the finance job uses to verify a valid 4700 user when a 4700 sign-on is received.

***NONE:** No user IDs are verified.

user-table-name: Specify the name of a user table that defines user IDs for the 4700.

JOB Parameter

Specifies the job name that is associated with the submitted finance job.

QFNCJOB: The job name is submitted as QFNCJOB.

job-name: Specify the job name that is associated with the submitted finance job.

JOBD Parameter

Specifies the name of the job description that is used by the finance job.

The possible library values are:

***LIBL:** The library list is used to locate the finance job.

***CURLIB:** The current library for the job is used to locate the finance job. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the finance job is located.

QFNC: The submitted finance job uses the job description QFNC.

job-description-name: Specify the name of a job description that is used by the finance job. (If no library name is given, the job description is found through the library list used by the job in which the SBMFNCJOB command is entered.)

MSGQ Parameter

Specifies the name of the message queue, if any, to which messages are sent while the finance job is running.

WRKSTN: The finance messages are sent to the message queue of the work station from which the finance job was submitted.

***NONE:** No finance messages are sent to a message queue.

The possible library values are:

***LIBL:** The library list is used to locate the message queue.

***CURLIB:** The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message queue is located.

message-queue-name: Specify the name of the message queue to which messages are sent. (If no library name is given, the library list of the job issuing the SBMFNCJOB command is used to find the queue.)

Examples

```
SBMFNCJOB  DEVTBL(DEVTBL1)  PGMTBL(PGMTBL1)  +
  USRTBL(USRTBL1)
```

This command causes batch job QFNCJOB to be submitted. The job communicates with all devices it acquires from device table DEVTBL1, allowing users whose user IDs are found in USRTBL1 to sign on the devices. Each transaction sent by the finance devices is processed by determining, in PGMTBL1, which application program must be called, then it calls that program.

The job description used by the finance job in this example is QFNC.*LIBL. Messages sent as a result of finance job are sent to the message queue of the work station from which the job was submitted.

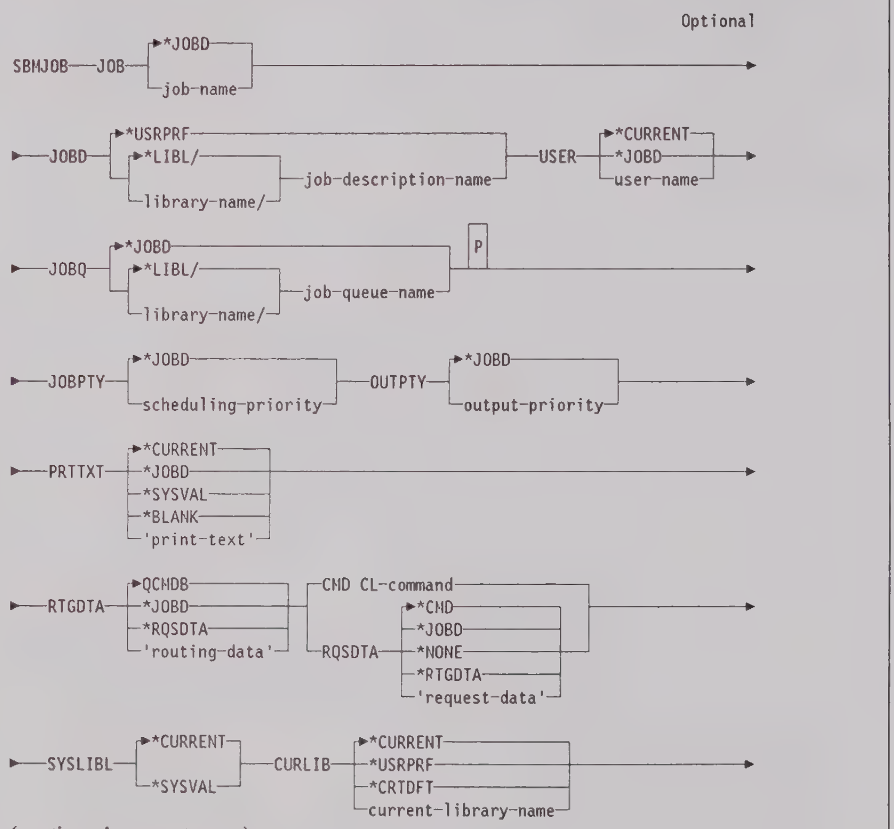
```
SBMFNCJOB  DEVTBL(DEVTBL2)  PGMTBL(PGMTBL2)  JOB(CTFJOB)  +
  JOBD(CTFJOBD) MSGQ(*NONE)
```

This command causes batch job CTFJOB to be submitted. CTFJOB runs under job description CTFJOBD and does not send messages to any work station message queue while running. No verification of user IDs is performed by the finance job.

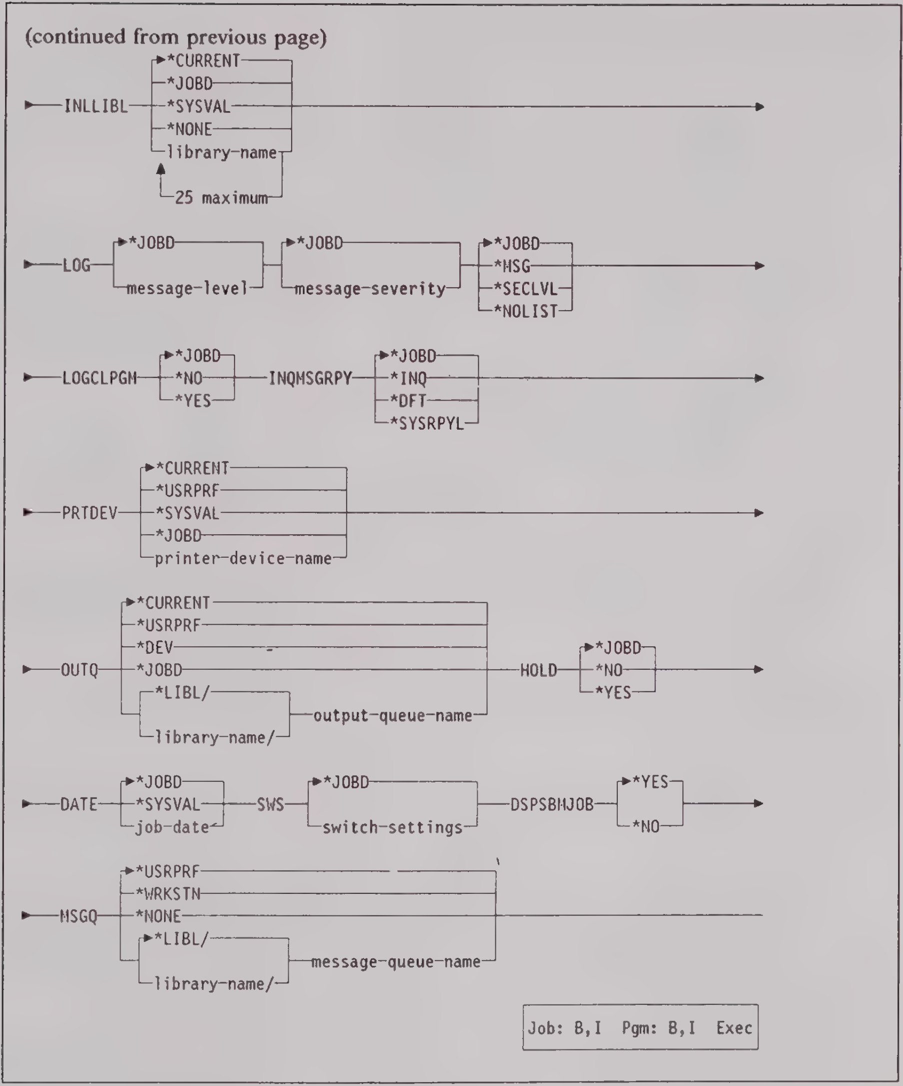
SBMJOB (Submit Job) Command

The Submit Job (SBMJOB) command allows a job that is running to submit another job to a job queue to be run later as a batch job. Only one element of request data can be placed on the new job's message queue. The request data can be a CL command if the routing entry used for the job specifies a CL command processing program.

Note: A job started by the SBMJOB command uses the accounting code of the job that submits the job. The accounting code specifications on the submitted jobs' JOBD and USRPRF parameters are ignored.



(continued on next page)



JOB Parameter

Specifies the job name associated with the submitted job as it is processed by the system.

***JOBD:** Specifies the simple name of the job description used with this job is the name of the job itself.

job-name: Specify the simple name of the job that is used while it is being processed by the system.

JOBD Parameter

Specifies the qualified name of the job description used with the job being submitted.

***USRPRF:** The job description in the user profile under which the submitted job runs is used as the job description of the submitted job.

The possible library values are:

***LIBL:** The library list is used to locate the job description name.

library-name: Specify the library where the job description name is located.

job-description-name: Specify the qualified name of the job description used for the job.

USER Parameter

Specifies the name of the user profile that is associated with the job being submitted. If USER(*RQD) is specified in the job description, ***JOBD** cannot be specified here; ***CURRENT** or *user-name* must be specified instead.

***CURRENT:** The same user profile used by the job that is currently running is used for the submitted job.

***JOBD:** The user profile named in the specified job description is used for the job being submitted.

user-name: Specify the name of the user profile that is used for the job being submitted. You must be authorized to the user-name; user-name must be authorized to the JOBD.

JOBQ Parameter
Specifies the name of the job queue where the submitted job is placed.

*JOBD: The submitted job is placed in the job queue named in the specified job description.

The possible library values are:

*LIBL: The library list is used to locate the job queue name.

library-name: Specify the library where the job queue name is located.

job-queue-name: Specify the qualified name of the job queue where the submitted job is placed.

JOBPTY Parameter
Specifies the scheduling priority for the submitted job. Valid values are 1 through 9, where 1 is the highest priority and 9 is the lowest. (For an expanded description of the JOBPTY parameter, see "Scheduling Priority Parameters" in Appendix A, "Expanded Parameter Descriptions.")

*JOBD: Specifies the scheduling priority specified in the job description is used for the job.

scheduling-priority: Specify a value, 1 through 9, for the scheduling priority for the job.

OUTPTY Parameter
Specifies the output priority for spooled output files that are produced by the submitted job. The highest priority is 1 and the lowest is 9. (For an expanded description of the OUTPTY parameter, see "Scheduling Priority Parameters" in Appendix A, "Expanded Parameter Descriptions.")

*JOBD: Specifies the output priority specified in the job description is used for the job.

output-priority: Specifies a value, 1 through 9, for the priority that the submitted job's output files has.

PRTTXT Parameter
Specifies up to 30 characters of text that is printed at the bottom of each page of printed output and on separator pages. (For an expanded description of the PRTTXT parameter, see Appendix A, "Expanded Parameter Descriptions.")

*CURRENT: Specifies the same print text of the submitting job is used.

***JOBD:** Specifies the value in the job description is used for this job.

***SYSVAL:** Specifies the system value (QPRTTXT) is used.

***BLANK:** Specifies no text is printed.

'print-text': Specify the character string that is printed at the bottom of each page. A maximum of 30 characters can be entered and must be enclosed in apostrophes.

RTGDTA Parameter
Specifies the routing data used with this job description to start jobs. The routing data is used to determine the routing entry (in the subsystem description) that identifies the program where the job runs.

*QCMDB: Specifies the routing data used by the IBM-supplied interactive batch subsystem QBATCH to route batch jobs to the IBM-supplied control language processor QCMD.

***JOBD:** Specifies the routing data used to start the routing step is in the job description used with the job.

***RQSDTA:** Specifies up to 80 characters of the request data specified in the RQSDTA parameter of this command is used as the routing data for the job.

'routing-data': Specify the character string that is used as the routing data for starting the job. A maximum of 80 characters can be entered (enclosed in apostrophes if necessary).

CMD CL Parameter
Specifies a command that runs in the batch job if the routing program used when this batch job is started is one of the IBM-supplied default routing programs (QCMD or QCL). Because this command is used for the request data, this parameter is mutually exclusive with the RQSDTA parameter. The command cannot be longer than 256 characters. You can use command prompting for the command you want to specify on this parameter if you are using the prompter for the SBMJOB command.

RQSDTA Parameter
Specifies the request data that is placed in the submitted job's message queue. For example, if RTGDTA (QCMDB) is specified, the IBM-supplied batch subsystem QBATCH is being used, and a CL command is supplied, it becomes a message that is read by the control language processor, QCMD.

*CMD: Specifies the input from the CMD parameter is placed in this job's message queue.

***JOBD:** Specifies the request data specified in the job description used by the job is placed in this job's message queue.

***NONE:** Specifies no request data is placed in the job's message queue.

***RTGDTA:** Specifies the routing data specified in the RTGDTA parameter of this command is placed as the last entry in the job's message queue.

'request-data': Specify the character string that is placed as the last entry in the submitted job's message queue. A maximum of 256 characters can

be entered (enclosed in apostrophes if necessary). When a CL command is entered, it must be enclosed in single apostrophes, and where apostrophes would normally be used *inside* the command, double apostrophes must be used instead.

SYSLIBL Parameter

Specifies the system portion of the library being used by the submitted job.

***CURRENT:** The system portion of the library list of the submitting job is used.

***SYSVAL:** The library list specified in the system value (QSYSLIBL) at the time the job is started is used for the submitted job.

CURLIB Parameter

Specifies the name of the current library associated with the job being run.

***CURRENT:** Specifies the current library being used by the job that is currently running is used for the submitted job.

***USRPRF:** Specifies the current library in the user profile under which the submitted job runs is used as the current library for the submitted job.

***CRTDFT:** Specifies there is no current library for the submitted job. If any objects are created in the current library, QGPL is used as the default current library.

current-library-name: Specify the name of a library used as the current library of the submitted job.

INLLIBL Parameter

Specifies the user portion of the first library that is used by the submitted job to search for any object names that were specified without a library qualifier. This does not include the system portion of the library list.

***CURRENT:** Specifies the user portion of the library list being used by the job that is currently running is used for the submitted job.

***JOBD:** Specifies the library list specified in the job description used with the job is used as the first library list for the job.

***SYSVAL:** Specifies the system default library list (QUSRLIBL) is used by the job.

***NONE:** Specifies the user portion of the first library list is empty; only the system portion is used.

library-name: Specify the names of one or more libraries that are in the user portion of the library list and are used by the submitted job. No more than 25 names can be specified; the libraries are searched in the same order as they are listed here.

LOG Parameter

Specifies the message logging values used by the submitted job. They determine the amount and type of information that is logged in the job log. The LOG parameter is made up of a list of three values: the message (or logging) level, the message severity, and the level of message text. If no values are specified for the LOG parameter, the values specified in the job description used with the job are used.

***JOBD:** Specifies the value specified for message logging in the job description is used for the submitted job.

message-level: Specify a value, 0 through 4, that specifies the message logging level used for the submitted job's messages. (For additional information on the message levels, refer to "Message Level" under the CRTJOBD command's LOG parameter.)

message-severity: Specify a value, 00 through 99, that specifies the lowest severity level that logs an error message in the job's log. (For an expanded description of severity codes, see the SEV parameter in Appendix A, "Expanded Parameter Descriptions.")

***MSG:** Specifies only message text is written to the job's log or displayed to the user.

***SECLVL:** Specifies both the message text and help text of the error message are written to the job's log or displayed to the user.

***NOLIST:** Specifies no job log is produced if the job ends normally. If the job ends abnormally (if the end of job code is 20 or higher), a job log is produced. The messages appearing in the job's log contains both message text and help text.

LOGCLPGM Parameter

Specifies if the loggable commands that have run in a control language program are logged to the job log by way of the CL program's message queue. This parameter sets the status of the job's logging flag. If *NO is specified, the logging flag status is *off* and CL commands are not logged. If *YES is specified and the LOG (*JOB) value is specified in the Create CL Program (CRTCLPGM) command, all commands in the CL program that can be logged are logged to the job log.

For more information on request logging, refer to the LOG parameter in the CRTCLPGM command description.

***JOBD:** Specifies the value specified in the job description is used.

***NO:** Specifies the commands in a CL program are not logged to the job log.

***YES:** Specifies that commands in a CL program are logged to the job log.

INQMSGRPY Parameter

Specifies the way that predefined messages are answered (predefined messages that are sent as a result running this job). You can specify that no change is made in the way that predefined messages are answered, that all inquiry messages require a reply, that a default reply be issued, or that the system reply list is checked for a matching reply as each predefined inquiry message is sent. Refer to the ADDRPYLE (Add Reply List Entry) command for more information.

***JOBD:** Specifies the inquiry message reply control specified in the job description used with this job is used.

***INQ:** Specifies a reply is required by the receiver of the inquiry message for all inquiry messages that occur when this command is run.

***DFT:** Specifies the default message reply is used to answer any inquiry messages issued when this command is run.

***SYSRPYL:** Specifies the system reply list is checked to see if there is an entry for any inquiry message that is issued as a result of running this job and has a message identifier and comparison data (if any) that match the inquiry message identifier and message data. If a match occurs, the reply value in that entry is used. If no entry exists for that message, a reply is required.

PRTDEV Parameter

Specifies the name of the printer device used as the default printer device for this job.

***CURRENT:** Specifies the same printer device being used by the job that is currently running is used by the submitted job.

***USRPRF:** Specifies the printer device name in the user profile under which the submitted job runs is used as the printer device for this job. The printer device name is obtained from the profile when this command is run.

***SYSVAL:** Specifies the printer device name in the system value QPRTDEV when this command is run is used for the submitted job.

***JOBD:** Specifies the printer device name specified in the job description is used for the submitted job.

printer-device-name: Specify the name of the printer device used for the submitted job.

OUTQ Parameter

Specifies the name of the default output queue that is used for spooled output produced by the submitted job.

***CURRENT:** The output queue used by the job that is currently running is used for the submitted job.

***USRPRF:** Specifies the output queue in the user profile under which the submitted job runs is used as the output queue for the submitted job. The output queue name is obtained from the profile when this command is run.

***DEV:** Specifies the output queue associated with the specified printer device is used.

***JOBD:** Specifies the output queue named in the job description used with the submitted job is the job's default output queue.

The possible library values are:

***LIBL:** The library list is used to locate the output queue name.

library-name: Specify the library where the output queue name is located.

output-queue-name: Specify the qualified name of the output queue that is used as the default output queue by the submitted job.

HOLD Parameter

Specifies if the submitted job is held at the time that it is put on the job queue. A job placed on the job queue in the hold state is held until it is released by the Release Job (RLSJOB) command, canceled by the End Job (ENDJOB) command, or removed from the job queue by the Clear Job Queue (CLRJOBQ) command. If the job is not run before OS/400 is ended, the job queue can be cleared (and the job ended) when OS/400 is started again.

***JOBD:** Specifies the value specified in the job description determines if the job is held when it is put on the job queue.

***NO:** Specifies the job is not held when it is put on the job queue.

***YES:** Specifies the job is held when it is put on the job queue, and held until it is released by a RLSJOB command or ended by a ENDJOB command.

DATE Parameter

Specifies the date that is assigned to the submitted job when it is started.

***JOBD:** Specifies the date specified in the job description is used as the job date.

***SYSVAL:** Specifies the value in the QDATE system value at the time the job is started is used as the job date.

job-date: Specify the value that is used as the job date when the job is started; the date must be in the format specified by the system value QDATFMT. (Refer to the *CL Programmer's Guide* for the formats.)

SWS Parameter

Specifies the first settings for a group of eight job switches to be used with the submitted job. These switches can be set or tested in a CL program and used to control the flow of the program. For example, if a certain switch is on, another program could be called. The job switches may also be valid in other high-level language programs. Only zeros (off) and ones (on) can be specified in the 8-digit character string.

***JOBD:** Specifies the value specified in the job description is the first settings for the job's switches.

switch-settings: Specify any combination of eight zeros and ones that is used as the first switch setting for the submitted job.

DSPSBMJOB Parameter

Specifies if the job being submitted is allowed to be shown on the Submitted Jobs Display. Any submitted job of the type specified by the SBMFROM parameter of the Work with Submit Job (WRKSBMJOB) command can be shown if the job is not prevented by this parameter.

***YES:** This job can be shown by the WRKSBMJOB command.

***NO:** This job is not shown on any display produced by the WRKSBMJOB command.

MSGQ Parameter

Specifies the message queue, if any, to which a completion message is sent when the submitted job has completed running, either normally or abnormally. If a job ends abnormally, the help text of the completion message sent specifies the possible causes.

***USRPRF:** Specifies a completion message is sent to the message queue specified in the user profile of the submitter.

***WRKSTN:** Specifies a completion message is sent to the work station message queue of the work station from which the job was submitted. If the job is submitted by a batch job, no completion message is sent.

***NONE:** Specifies no completion message is sent.

The possible library values are:

***LIBL:** The library list is used to locate the message queue name.

library-name: Specify the library where the message queue name is located.

message-queue-name: Specify the qualified name of the message queue where the completion message is to be sent.

Examples

```
SBMJOB  JOB(SPECIAL)  JOBD(MYLIB/MYJOBD)  CMD(CALL MYPROG)
```

This command causes the job named SPECIAL to be submitted. Most of the attributes for the job are taken from the job description MYJOBD or the job that is currently running, except for the command. The CALL command is placed on the submitted job′s message queue so that the program MYPROG can be called and run later.

```
SBMJOB  JOB(PAYROLL)  JOBD(PAYROLL)  INQMSGRPY(*RQD)
```

This command submits a job named PAYROLL to the system. All the information needed for this job (such as the job queue and routing data but not the inquiry message control value) is contained in the job description PAYROLL or the job that is currently running. The library list in effect for the job issuing this command is used to find the job description. All inquiry messages sent during running of this job requires the receiver of the inquiry message to reply.

```
SBMJOB  JOBD(*USRPRF)  JOB(COPY12)  JOBQ(NIGHTQ)  CMD(CPYF FILEA FILEB)
```

This command submits the job COPY12, which uses the job description in the user profile of the submitting job to the job queue NIGHTQ. The CMD parameter provides the CL command necessary for the job to run. A command such as this might be used to copy the file at night while the system is unattended.

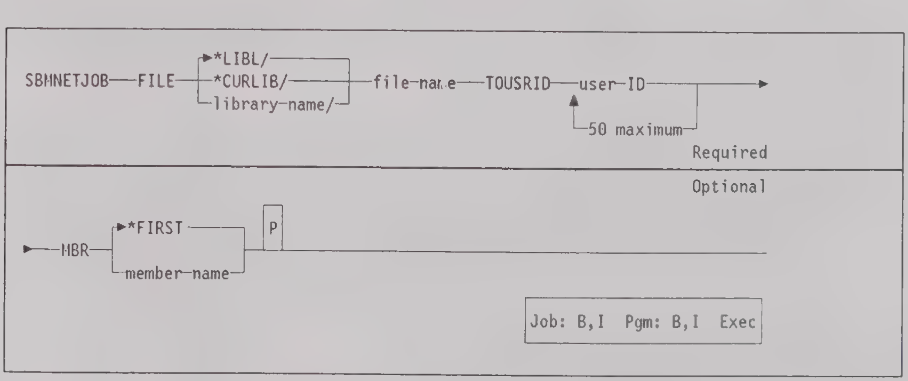
SBMNETJOB (Submit Network Job) Command

The Submit Network Job (SBMNETJOB) command sends a job stream to another system user on the SNADS network. (The job stream is sent to another user where it can be filed, submitted, or rejected.) When the job stream arrives, its placement is governed by the job action (JOBACN) network attribute. If the value of JOBACT is *SEARCH, the entry in the network job table at the receiving system is used to determine the action taken. (The network job table entry is determined by the value specified on the ACTION parameter on the Add Network Job Entry (ADDNETJOBE) or Change Network Job Entry (CHGNETJOBE) commands.) At the receiving system, the job may be submitted immediately, filed for placement by the receiving user, or rejected.

When the job stream arrives at the destination system, a message is sent to both the recipient of the job stream as well as the originator of the job stream stating that the job stream arrived.

This command can only be used to send a batch job stream to a user on a remote system.

Restrictions: (1) To use this command, you must have object operational and read authority to the file that is submitted, and for the library that contains the file. (2) You must be enrolled in the system distribution directory to use this command. (For information on enrolling in the system distribution directory, see the *CL Programmer's Guide*.) (3) If the job action (JOBACN) network attribute on the receiving system is set to *SEARCH, there must be an entry for you in the network job table on the receiving system. The entry in this table specifies a user profile on the receiving system that is used to verify that you are authorized to submit the job on that system. The user profile on the receiving system must be authorized to use the job queue(s), and must have object operational authority for the job description(s) specified by the JOB command(s) in the job stream. (4) The file that is submitted cannot contain more than approximately 16 million bytes of data. (5) The internal value for a node identifier may differ from the characters shown by the SBMNETJOB command depending on the type of work station (language) you are using. If the byte-string value specified for the TOUSRID command parameter does not match the rules for an internal node identifier value or if it does not match the internal value for any defined node (ignoring case differences), an error may be reported.



FILE Parameter

Specifies the name of the physical file containing the job stream that is sent.

The possible library values are:

***LIBL:** The library list is used to locate the file name.

***CURLIB:** The current library for the job is used to locate the file name. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the file name is located.

TOUSRID Parameter

Specifies the two-part user ID of one or more users to whom the job stream is submitted, or the name of one or more distribution lists containing the two-part user IDs of one or more users to whom the file is to be sent. A combination of both user IDs and distribution lists can be specified on the same command. Each user ID or distribution list is specified as a two-part name, and both parts are required.

Notes:

1. Depending on the type of work station you are using, the internal value for a user identifier may differ from the characters shown by the Display Directory (DSPDIR) command. If the byte-string value specified for the TOUSRID parameter does not match the rules for an internal user identifier value, or if it does not match the internal value for any enrolled user, an error may be reported.
2. The user specified in this parameter, or in the distribution list, must be a remote user. The SBMNETJOB command cannot be used to send batch job streams to local users.

MBR Parameter
Specifies the member that is sent from the file.

***FIRST:** The first member (in creation order) in the file is submitted.

member-name: Specify the name of the file member that is submitted.

Example

```
SBMNETJOB FILE(JONES) TOUSRID(WEEKLY PAYROLL) MBR(SMITH)
```

This command sends the batch job streams contained in member SMITH of file JONES to user ID WEEKLY PAYROLL.

SBMRJEJOB (Submit RJE Job) Command

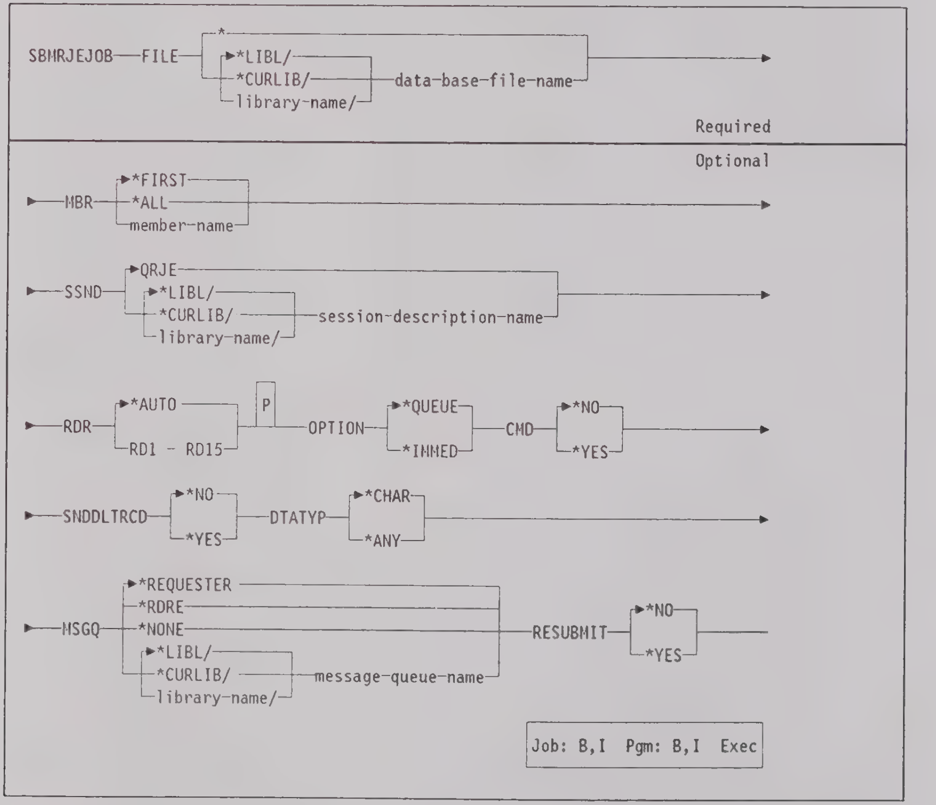
The Submit RJE Job (SBMRJEJOB) command sends an RJE remote job input stream to the host system.

Through the OPTION parameter of this command, an RJE reader job is either sent to the host system immediately, or placed on an RJE reader job queue to send to the host system as a batch job.

Through the FILE parameter (*), you can start an RJE interactive reader on a work station for direct reader input.

Restrictions: To use this command, you must have:

- Object operational authority to the session description.
- Object operational and read authority to the input data base file and any files named on embedded READFILE control statements.
- Read authority to the library where the session description and data base file are stored.



FILE Parameter

Specifies the qualified source of the remote job input stream submitted.

*: The remote job input stream is from the work station where the SBMRJEJOB command was sent.

An interactive input stream sends data one record at a time (each time Enter is pressed). The READFILE control statement embeds the data files and commands in the input stream. The EOF control statement is used to end the input stream.

No other jobs are submitted to the reader. If this value is specified, OPTION(*IMMED) and CMD(*YES) must also be specified.

In the systems network architecture (SNA) environment, at least two logical unit (LU) sessions are available when this option is entered.

The possible library values are:

***LIBL:** The library list is used to locate the file.

***CURLIB:** The current library for the job is used to locate the file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the file is located.

data-base-file-name: Specify the name of the data base file that contains the remote job input stream submitted.

You cannot override the file parameter with OVRDBF MBR(*ALL). This method results in RJE sending you an error message. You have the option of canceling this job or continuing. To format the members of a data base file, use the *ALL option after the library name.

MBR Parameter

Specifies the data base file member sent to the host system. This parameter applies only if you specified a data base file in the FILE parameter.

***FIRST:** Specifies the first member in the data base file.

***ALL:** Specifies all members in the data base file.

member-name: Specify the member name of the data base file. If the member does not exist, an inquiry message is sent to the RJE message queue.

SSND Parameter

Specifies the qualified name of the session description that sends the RJE job to the host system.

***QRJE:** QRJE is the session description name.

The possible library values are:

***LIBL:** The library list is used to locate the session description.

***CURLIB:** The current library for the job is used to locate the session description. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the session description is located.

session-description-name: Specify the name of the session description.

RDR Parameter

Specifies the RJE reader that is associated with the RJE job submitted.

***AUTO:** An RJE reader input stream available at the time this command runs.

RD1 - RD15: Specify a value from 1-15 for the specified reader, for example, RD1.

OPTION Parameter

Specifies if the RJE reader job is sent to an RJE reader job queue or started immediately.

***QUEUE:** The job is submitted to the job queue associated with the reader that you specified in the RDR parameter. The RJE session does not have to be active. The RJE reader job queue is released when the session is started and the STRRJERDR command is sent.

***IMMED:** The job starts immediately, and control is not returned to the work station until the job is complete. The RJE session must be active. The specified RJE reader must be started and not in use.

CMD Parameter

Specifies if the file specified in the FILE parameter is searched for embedded RJE control statements (READFILE or EOF).

***NO:** The file is not searched.

***YES:** The file is searched.

SNDDLTRCD Parameter

Specifies if blank records are sent to the host system for deleted records that are encountered in the data base file member.

***NO:** Blank records are not sent.

***YES:** Blank records are sent.

Files created through the data base join function are submitted. Deleted records are recognized as the primary part of the logically joined file.

DTATYP Parameter

Specifies if the file specified in the FILE parameter contains data characters less than X'40'.

***CHAR:** The file does not contain any data characters.

***ANY:** The file contains data characters.

MSGQ Parameter

Specifies the qualified name of the user message queue where messages for the RJE reader are recorded.

Note: Messages for RJE readers are recorded in the RJE message queue associated with the RJE session. The RJE message queue name depends on the name specified in the MSGQ parameter in the Create Session Description (CRTSSND) or Change Session Description (CHGSSND) command. If inquiry messages are sent by RJE, they are sent to the user message queue (if specified) where they must receive a response.

***REQUESTER:** Specifies the requester's work station message queue and the RJE message queue.

If this command is run in a batch job, *REQUESTER is treated as *NONE (inquiry messages cannot be sent to a batch job log).

***RDRE:** The message queue name is retrieved from the session-description reader entry associated with this RJE reader function.

***NONE:** A user message queue is not used.

The possible library values are:

LIBL: The library list is used to locate the message queue.

***CURLIB:** The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message queue is located.

message-queue-name: Specify the name of the user message.

RESUBMIT Parameter
Specifies if jobs in the queue are resubmitted if they are ended abnormally.

Note: The job sequence of each reader is preserved unless the jobs were submitted to the *AUTO reader. To use this parameter, a job submitted by CL commands must have a job priority (JOBPTY) of 6, and the user profile that submits RJE reader jobs must have the priority limit (PTYLMT) parameter set from 1 to 5, inclusive. The MAXACT parameter of the reader job queue entries in the RJE subsystem description must be set to 1. A reader job submitted with this parameter is canceled directly from the job queue. Other commands, such as CNLJOB OPTION(*IMMED), cause the job to be resubmitted.

***NO:** Files are not resubmitted.

***YES:** Files are resubmitted.

Example

```
SBMRJEJOB  FILE(USERLIB/CL)  MBR(*ALL)  SSND(USERLIB/RJE)  +
  RDR(*AUTO)  OPTION(*QUEUE)  CMD(*YES)  MSGQ(MEDLIB/BROWN)
```

This command sends data from a data base file named CL in library USERLIB to the host system. The job is submitted to an RJE reader job queue. The job queue is specified in the *AUTO reader entry in the session description named RJE in library USERLIB. The members in the data base file consist of command files. They are checked for READFILE and EOF control statements.

RJE messages associated with RD1 (reader 1) are written to a user message queue named BROWN in library MEDLIB.

SBMRMTCMD (Submit Remote Command) Command

The Submit Remote Command (SBMRMTCMD) command submits a command via Distributed Data Management (DDM) for running on the target system specified by a DDM file. The values for the DEV and MODE parameters in the DDM file determine the communications line used, and thus indirectly identify the target system that receives the submitted command.

This command is used to send CL commands (and only CL) to another AS/400, and is valid only when the target system is an AS/400. It cannot be used to send non-AS/400 commands; for example, OCL commands cannot be sent to a target System/36.

The primary purpose of this command is to allow a source system user or program to perform file management operations and file authorization activities on files located on a target AS/400. (More information on file management operations is in the *DDM User's Guide*.) If the source system program or user has the correct authority to do so, the following actions are examples of what is performed on remote files using the SBMRMTCMD command:

- Create or delete physical, logical, device, or source files
- Grant or revoke object authority to remote files
- Check, rename, or move files or other objects
- Save or restore files or other objects

Although the command is used to do many things with files or objects, some are not as useful as others. For example, you could use it to display the file descriptions or field attributes of remote files, or to dump files or other objects, but the output results remain at the target system. To display remote file descriptions and field attributes at the source system, use the DSPFD and DSPFFD commands, specifying SYSTEM(*RMT).

A secondary purpose of this command allows a user to perform nonfile operations, such as creating a message queue, or to submit user-written commands for running on the target system.

Restrictions: (1) Although remote file processing is synchronous in the user process, which includes two separate jobs (one running on each system), file processing on the target system operates independently of the source system. Commands such as OVRDBF, OVRMSGF, and DLTOVR that are dependent on a specific recursion level or request level may *not* function as expected. (2) Output (such as spooled files) generated by a submitted command exists only on the target system. The output is *not* sent back to the source system. (3) Some types of CL commands should *not* be submitted to a target AS/400. The following are examples of types that are *not* the intended purpose of the SBMRMTCMD command and that may produce undesirable results.

- All OVRxxxF commands that refer to device files, communications files, message files, or save files.
- The DSPxxxx commands should not be submitted because the output results remain at the target system.
- Commands that are valid only in an interactive environment, like EDTSRC or ENTPS.
- Job-related commands like RRTJOB that are used to control a target system's job. The CHGJOB command, however, *can* be used.
- Commands that are used to service programs, like SRVJOB, TRCJOB, TRCINT, or DMPJOB.
- Commands that may cause inquiry messages to be sent to the system operator, like STRPRTWTR or CPYTODKT. Pass-through can be used instead.

(4) Translation is not performed for any *impromptu* messages caused by target system errors, because they are not stored on the system; the text for an impromptu message is sent directly to the source system displayed. The message identifier of all other message types generated on the remote system is sent back to the source system. If the message text that exists for the message identifier on the source system has been translated, it will then be displayed. (5) A maximum of 10 messages, generated during the running of a submitted command, can be sent by the target system to the source system. If more than 10 messages

are generated, an additional *informational* message is sent that indicates that the messages exist in the job log for the target job on the target system. If one of those messages is an *escape* message, the first nine messages of other types are sent, followed by the informational message and the escape message.

```
                                                                Required
SBMRMTCMD───CMD 'command'──────────────────────────────────────────────►

              ┌►*LIBL/──────────┐
►───DDMFILE───┼─*CURLIB/────────┼───DDM file-name──────────────────────
              └─library-name/───┘

                                                ┌─────────────────────────┐
                                                │Job: B,I  Pgm: B,I  Exec │
                                                └─────────────────────────┘
```

CMD Parameter

Specifies a character string of up to 2000 characters that represents a command that is run on the target system. The command must be allowed in both *BATCH and *EXEC (QCAEXEC) environments on the target AS/400.

The command must be enclosed in apostrophes if it contains embedded blanks or special characters.

Note: The normal rule of pairing apostrophes in quoted strings on the local system must be *doubled* when the same string is submitted to a remote system on this CMD parameter; this is required because you are coding a quoted string within another quoted string. Therefore, when this parameter is being coded, wherever a single apostrophe would normally be paired with another apostrophe, *each occurrence in the outside set of apostrophes must be doubled* to produce the same results at the target system. (This is shown in the third and fourth examples of the following section.)

DDMFILE Parameter

Specifies the qualified name of the DDM file that is used to submit the command to the target system. (If no library qualifier is given, *LIBL is used to find the file.) The DDM file is used only to determine the remote location representing the target system. The remote file name associated with the DDM file is ignored by the SBMRMTCMD command.

The possible library values are:

<u>*LIBL:</u> The library list is used to locate the DDM file.

***CURLIB:** The current library for the job is used to locate the DDM file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the DDM file is located.

Examples

```
SBMRMTCMD  CMD('DLTF LIBX/FRED')  DDMFILE(DENVER)
```

This command deletes the file named FRED in library LIBX on the target system that is associated with the DDM file named DENVER.

```
SBMRMTCMD  CMD('CRTPF  SALES/CAR  SRCFILE(QGPL/QDDSSRC)  +
  SRCMBR(MASTER)')  DDMFILE(DENVER)
```

This command creates the physical file CAR in library SALES using the data description specifications (DDS) in the source file QDDSSRC and source member named MASTER in the QGPL library. The DDS must already exist on the target system identified by the DDM file named DENVER in the target job's library list.

```
SBMRMTCMD  CMD('CHGDDMF  FILE(LIBX/STANLEY)  +
  TEXT(''Don''''t forget to pair apostrophes.'')')  +
  DDMFILE(SMITH)
```

This command changes the text in the description of the DDM file named STANLEY stored in library LIBX. Because the submitted command requires an outside set of single apostrophes (for the CMD parameter), each single or double apostrophe normally required in the TEXT parameter for *local* system processing must be doubled again for *remote* system processing. The coding above produces a single apostrophe in the text when it is displayed or printed on the remote system.

```
SBMRMTCMD CMD('CRTDDMF FILE(SALES/MONTHLY)  +
  RMTFILE(*NONSTD ''CAR.SALES(JULY)''  +
  RMTLOCNAME(DALLAS)') DDMFILE(CHICAGO)
```

This command creates (on the target system identified by the information in the DDM file named CHICAGO) another DDM file named MONTHLY. The new DDM file is stored in a library named SALES on the CHICAGO system. The new DDM file on the CHICAGO system is used to access a file and *member* on a different system named DALLAS. The accessed file is named SALES/CAR and the member name in the file is JULY.

Note that this CRTDDMF command string contains *three* sets of single apostrophes: one set to enclose the entire command being submitted, and a double set to enclose the file and member named in the RMTFILE parameter. This is how any IBM AS/400 file *member* name must be specified on the SBMRMTCMD command, because of the parentheses needed to enclose the member name.

```
SBMRMTCMD  CMD('RPLLIBL  LIBL(QGPL QTEMP SALES EVANS)')  +
  DDMFILE(EVANS)
```

This command replaces the user's portion of the library list being used by the target job associated with the DDM file named EVANS, which is being used by the source job in which this SBMRMTCMD command is being submitted. In that source job, if there are other open DDM files that specify the same device and mode, this library list is used for them also.

Additional Considerations

Override example: The DDMFILE parameter is used to determine which target system the command should be sent to. Overrides that apply to the DDM file (not the remote file) are taken into account for this function. For example, if a file override was in effect for a DDM file because of the following commands, which override FILEA with FILEX, then the target system that the DLTF command is sent to is the one associated with the remote location values specified in FILEX (the values point to the DENVER system, in this case).

```
CRTDDMF  FILE(SRCLIB/FILEA)  RMTFILE(SALES/CAR)  +
  RMTLOCNAME(CHICAGO)

CRTDDMF  FILE(SRCLIB/FILEX)  RMTFILE(SALES/CAR)  +
  RMTLOCNAME(DENVER)

OVRDBF   FILE(FILEA)  TOFILE(SRCLIB/FILEX)  +
  SBMRMTCMD  CMD('DLTF RMTLIB/FRED')  DDMFILE(SRCLIB/FILEA)
```

Distributed Data Management conversations: When a SBMRMTCMD command runs on the target system, it has a target system job associated with it. Successive SBMRMTCMD commands submitted using the same DDM file and DDM conversation may run in the same or different target system jobs, depending on the DDMCNV job attribute value. The DDMCNV job attribute value determines if the DDM conversation is dropped (DDMCNV(*DROP)) or remains active (DDMCNV(*KEEP)) when the submitted function has completed. If the conversation is dropped, the next SBMRMTCMD command runs using a different target job.

Command syntax checking: The syntax of the command character string being submitted by the CMD parameter is not checked by the source system. In the case of a user-defined command, for example, the command definition object may or may not exist on the source system.

Command running results: Because the submitted command runs as part of the target system's job, the attributes of that job (such as the library search list, user profile, wait times, and running priority) may cause a different result than if the command were run locally.

Error message handling:

- For errors detected by the target system when processing the command that was submitted, the source system attempts to send to the source system user the same error information that the target system sent. However, if the source system does not have an equivalent message for the one sent by the target system, source system default message text (stating that the message does not exist on the source system) is sent to the source system user, along with the message identifier and message type of the actual target system message. The target system messages can be viewed on the source system by using pass-through and either the DSPJOB or DSPJOBLOG command.
- If the SBMRMTCMD command is used to call a CL program on the target system, any unmonitored escape message generated by the program is changed into an inquiry message and is sent to the system operator. If you don't want the target system operator to have to respond to this inquiry message before the job can continue, you can refer to the following commands and do either of the following on the target system:
  - If you want to specify a default reply for a specific job, the INQMSGRPY parameter on either the CRTJOBD or CHGJOBD command can be used to specify either *DFT or *SYSRPYL in the job description for the target job.
  - If you want to specify a default reply message for a specific inquiry message, the ADDRPYLE command can be used (on the target system) to add an entry to the system-wide automatic message reply list for that message.

SETATNPGM (Set Attention Program) Command

The Set Attention Program (SETATNPGM) command causes the specified program to become the attention key handler at the current recursion level in the job running the command. If the ATTN key handler's status is SET(*ON), the specified program is called when the ATTN key is pressed. No parameters are passed to the ATTN key handler when it is called. The attention handling program runs in the same process with the same job attributes, overrides, and group authorities as the program that issued the SETATNPGM command. However, program adopted authority is not carried over.

This command is call-oriented; that is, a SETATNPGM command issued at one recursion level causes the attention handling program to be in effect at the current recursion level as well as deeper recursion levels, until another SETATNPGM command occurs which changes the attention handling program. The *Work Management Guide* has more information.

Pressing the ATTN key under the following conditions does not call the ATTN key handler (that is, it appears as if SET(*OFF) were the status of the ATTN key):

- When the input inhibited light at the work station is on
- At the system request menu or any of its options
- At the break message display and the display message display
- While OS/400 is already calling the attention handler
- Some OS/400 functions (3270 Emulation, 5250 Passthru) do not allow the ATTN key to call the attention handling program

Note: Caution is necessary when defining an attention handling program. Attention handling is similar to system request option 1 (transfer to secondary job), but there are special considerations for its use. When a transfer to secondary job occurs, processing continues in another job; however, when the ATTN key is pressed, the ATTN key handler runs in the same job. The program in progress when the ATTN key is pressed is not protected by any locks held. This means that the attention handler can get a lock on any object, even if a program it interrupted already had an exclusive lock. Be aware that read from invited devices operation could time-out during the time that the attention key handling program is running. Therefore, if a time-out were to complete in the program in progress while the attention key handling program is running, whatever action taken as a result of that time-out occurs upon return to the program in progress.

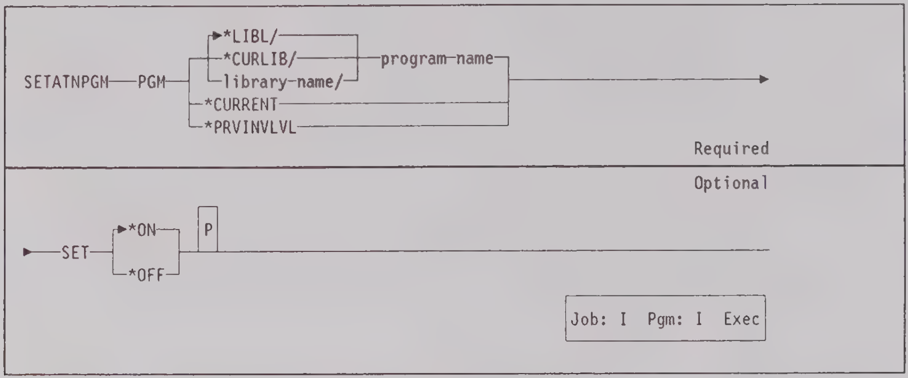
For example, if the WAITRCD value in the file is set to 60 seconds, the program is set to exit if a key is not pressed in 1 minute, and the attention key program is invoked and runs longer than 1 minute, the program exits when it returns from the attention key handler.

However, caution should be used because a check for data available is made before checking to see that the time-out completes. If a key is pressed immediately after leaving the attention key handler, data could be available which could complete the read from invited devices and the time-out is not checked. This could cause unexpected results.

Applications that perform Get-No-Wait operations can be interrupted at any point where the ATTN key is pressed. Attention handlers therefore should:

- Use simple functions like a menu which transfers to a group job or secondary job.
- Avoid referring to objects or functions that may be in use when the ATTN key is pressed.
- Avoid calling nonrecursive functions (many non-OS/400 functions) that are going to be interrupted by the ATTN key. (This is especially true of high-level languages and utilities, because they are not designed to be used recursively.)

Restriction: To use this command the program specified must exist and you must have object operational authority or one of the data authorities to it. You must also have read authority to the program's library.



PGM Parameter
Specifies the qualified name of the program to be the ATTN key handler at this recursion level.

The possible library values are:

***LIBL:** The library list is used to locate the program.

***CURLIB:** The current library for the job is used to locate the program. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the program is located.

***CURRENT:** The program name of the ATTN key handler currently in effect is used as the value of this parameter. (An error does not occur if an ATTN key handler is not currently in effect.)

***PRVINVLVL:** The ATTN key handler in effect at the previous recursion level is reinstated as the ATTN key handler at this recursion level. The SET parameter is not allowed if this special value is specified, because the SET status of the previous recursion level is also be reinstated. This option is used when a program has specified an attention program and wants to revert back to a previous level.

SET Parameter
Specifies if the ATTN key handler indicated in the program (PGM) parameter should be called when the ATTN key is pressed. This parameter is not allowed if PGM(*PRVINVLVL) is specified.

***ON:** The ATTN key handler specified in the program (PGM) parameter is called when the ATTN key is pressed.

***OFF:** The ATTN key handler specified in the program (PGM) parameter is not called when the ATTN key is pressed.

Examples

```
SETATNPGM PGM(QGPL/ATTN)  SET(*ON)
```

This command causes the program QGPL/ATTN to become the ATTN key handler. Because SET(*ON) is specified, the program is called when the ATTN key is pressed.

```
SETATNPGM  PGM(*CURRENT)  SET(*OFF)
```

The current attention handling program has its status changed to SET(*OFF). Because the status is SET(*OFF) when the ATTN key is pressed, the attention handling program is not called.

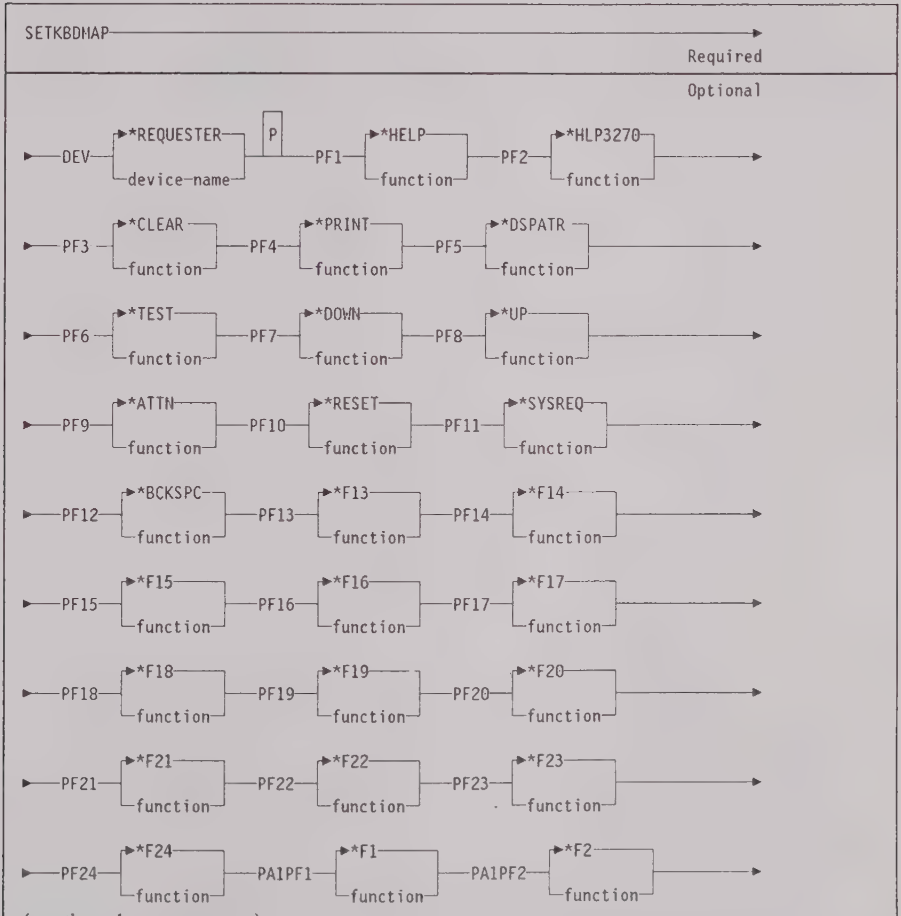
```
SETATNPGM  PGM(*PRVINVLVL)
```

The attention handling program and status that was in effect previous to the current one is reinstated at this recursion level. If no attention handler is in effect, after this command is run nothing happens when the ATTN key is pressed.

SETKBDMAP (Set Keyboard Map) Command

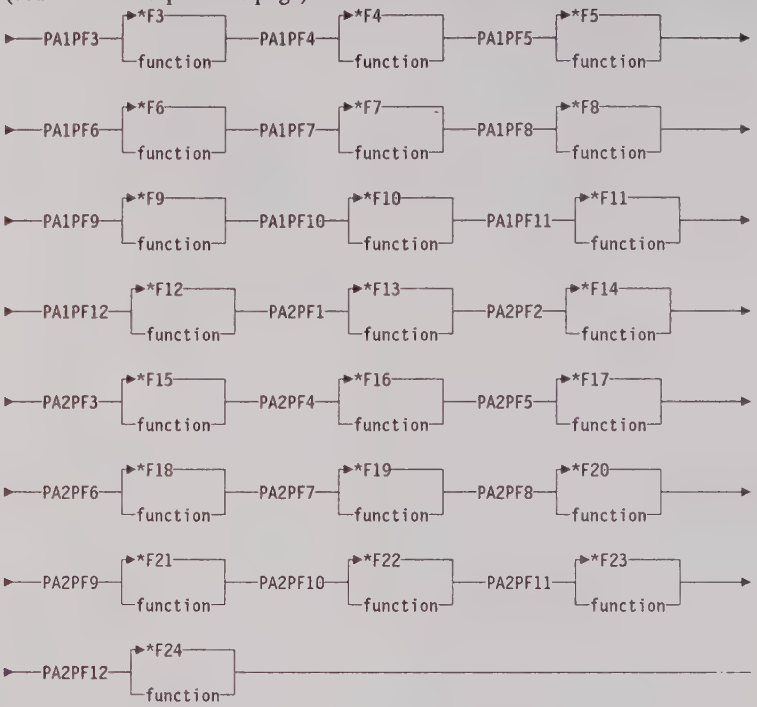
The Set Keyboard Map (SETKBDMAP) command allows the user to override the PA (Program Attention) and PF (Program Function) key assignment defaults provided by the AS/400 system.

This command assigns the specified F-to-PF map to the device where the command was entered (if it is a 3270 display station device) or to the 3270 display station specified if the user has authority to that device. More information on the user-assignment keyboard mapping is in the *Display Station User's Guide*.



(continued on next page)

(continued from previous page)



```
The following are allowable special values which are valid for all parameters:

*HELP      *TEST   *RESET    *F1 through *F24
*HLP3270   *DOWN   *SYSREQ
*PRINT     *UP     *BCKSPC
*DSPATR    *NONE   *ATTN
```

Job: B,I Pgm: B,I Exec

DEV Parameter

Specifies a valid 3270 display station that is assigned this keyboard mapping function.

__*REQUESTER:__ This mapping is assigned to the device where the command is entered.

device-name: Specify a device other than the device named in the *REQUESTER. You must have allocation rights to the specified device. In a program environment, you should either acquire or allocate the specified device before entering this command.

Key Sequence Parameter

Each valid key or key sequence is a separate parameter which, except for restrictions noted below, may be assigned any function. If you do not specify a function for a particular key or key sequence, the default value becomes the function for that key or key sequence. Following is a list of functions and restrictions:

*ATTN	Attention
*BCKSPC	Record Backspace
*CLEAR	Clear Screen
*DOWN	Roll Down
*DSPATR	Display Imbedded Attributes
*F1 – *F24	F1 through F24 Function Keys
*HELP	5250 Help
*HLP3270	3270 Help Text (Display Active Keyboard Map)
*NONE	No Assignment
*PRINT	Print Screen
*RESET	Error reset
*SAME	No Change
*SYSREQ	System Request
*TEST	Test Request
*UP	Roll up

Notes:

1. *HELP, *HLP3270, and *RESET must be assigned to PF1 through PF12, PA1/PF1 through PF12, or PA2 PF1 through PF12. Because these functions are required for 3270 display station device support to function properly, and not all keyboards have 24 PF keys, *HELP, *HLP3270, or *RESET may not be assigned to keys PF13 through PF24, unless

that function is also assigned to PF1 through PF12, PA1/PF1 through PF12, or PA2/PF1 through PF12.

2. It is recommended that *F1 and *SYSREQ be assigned to PF1 through PF12, PA1/PF1 through PF12, or PA2/PF1 through PF12.
3. The value *ATTN cannot be explicitly assigned to a 3270 remote attach display station. If the default value *ATTN is taken, the value *NONE is substituted. However, if the value *ATTN is explicitly chosen, a diagnostic message is sent.

Example

```
SETKBDMAP PF1(*F1)  PF2(*F2)  PF3(*F3)  PF4(*F4)  PF5(*HLP3270)  +
  PF9(*HELP)
```

This command reassigns your keyboard primarily for an application that makes frequent use of the 5250 CF keys F1, F2, F3, F4. All other PF key sequences are set to the default displayed on the command prompt. The above command is started in the program that started the application (thus tailoring the display station to whatever application is run).

SETPGMINF (Set Program Information) Command

The Set Program Information (SETPGMINF) command is used to place the names of all subprograms that are used at run time, and the names of all library information files, with the program specified in the ROOTPGM parameter. The command stores all the subprogram information that you specify and adds implicit subprograms for resolving the symbols in the specified subprograms.

You do not need to use this command if your program consists of only one compilation unit or the program calls only the NONPASCAL programs. For all other cases, you must use the SETPGMINF command before calling the program.

For example, if you have programs P1, P2, and P3, as follows:

```
program P1;
procedure F(const X :integer); external;

begin   (* main program begin *)
   F(1);
end;.

segment P2;
procedure G(var X :integer); external:

procedure F(const X :integer); external;
procedure F;
   var Y : integer;
  begin
     Y := X;
     writeln(Y);
     G(Y);
  end;.

segment P3;
procedure G(var X :integer); external;
procedure G;
  begin
     X := X + 1;
     writeln(X);
end;.
```

You can run the following SETPGMINF command and then run the program:

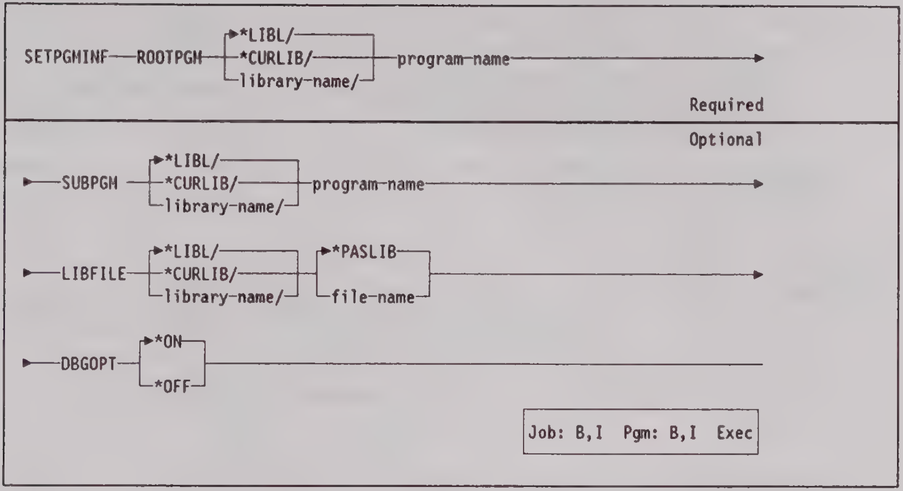
```
SETPGMINF ROOTPGM (P1) SUBPGM((P2) (P3)) LIBFILE((*PASLIB))
```

Subprogram information can be nested. For example, the above SETPGMINF command is equivalent to the following two SETPGMINF commands:

```
SETPGMINF ROOTPGM(P2) SUBPGM(P3) LIBFILE((*PASLIB))
SETPGMINF ROOTPGM(P1) SUBPGM(P2) LIBFILE((*PASLIB))
```

Nesting subprogram information in this way can be useful in organizing large applications. Note that any subprograms added as a result of a LIBFILE search are not eligible for nesting. For example, the following will not work even though the first SETPGMINF command resolves P3 with P2, because P3 was resolved by library information file MYFILE. Therefore, the second SETPGMINF command will not see P3.

```
EXTPGMINF PGM(P3) FILE(QTEMP/MYFILE) CRTFILE(*YES)
SETPGMINF ROOTPGM(P2) LIBFILE((*PASLIB) (MYFILE))
SETPGMINF ROOTPGM(P1) SUBPGM(P2) LIBFILE((*PASLIB))
```



ROOTPGM Parameter

Specifies the qualified name of the program that is called to establish the environment.

When *root environment* is to be established, ROOTPGM is a program unit, that is the program that contains the main entry point. When *main* or *reentrant* environment is to be established (by using the QPXXCALL procedure), the root program is the segment unit that is called as if it were the main program.

The possible library values are:

__*LIBL:__ The library list is used to locate the program.

***CURLIB:** The current library for the job is used to locate the program. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the program is located.

SUBPGM Parameter

Specifies a list of programs to be included during run-time. These programs can either be Pascal or NONPASCAL programs. If a program containing a main entry point is specified in the subprogram, it is handled like a NONPASCAL program. NONPASCAL programs do not need to be specified as subprograms.

The possible library values are:

__*LIBL:__ The library list is used to locate the subprogram.

***CURLIB:** The current library for the job is used to locate the subprogram. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the program is located.

The dynamic loader searches for subprograms at run time according to the library specifications used in the SUBPGM parameters. For example, if *LIBL is used, the dynamic loader searches the library list. However, if a library name or*CURLIB is used, the dynamic loader uses the actual library name specified in the SUBPGM parameter, or the name of the current library that is active when you run the SETPGMINF command.

LIBFILE Parameter

Specifies a list of library information files that contain information to resolve external references at run time. After all of the linkage information supplied by the root program and subprograms are processed, any remaining unresolved external identifiers are searched for in the specified library information files (LIBFILE). A library information file is created and updated by the Extract Program Information (EXTPGMINF) command. A library information file contains a mapping of external symbols to the programs where they are defined. SETPGMINF uses this information to add subprograms to the list explicitly specified in the SUBPGM parameter. The library files are searched in the order that they are specified.

The possible library values are:

__*LIBL:__ The library list is used to locate the library information files.

***CURLIB:** The current library for the job is used to locate the library information files. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the file is located.

The dynamic loader searches for library information files at run time according to the library specifications used in the LIBFILE parameter. For example, if *LIBL is used, the dynamic loader will search the library list. However, if a library name or *CURLIB is used, the dynamic loader will use the actual library name specified in the LIBFILE parameter or the name of the current library that is active when you run the SETPGMINF command.

***PASLIB:** Specifies the name of the Pascal run time library file will be used.

file-name: Specify the name of the library information file.

DBGOPT Parameter

Specifies debug attributes given to the program.

***ON:** The debug option is selected.

***OFF:** The debug option is not selected. This option overrides the debug mode of a job.

Example

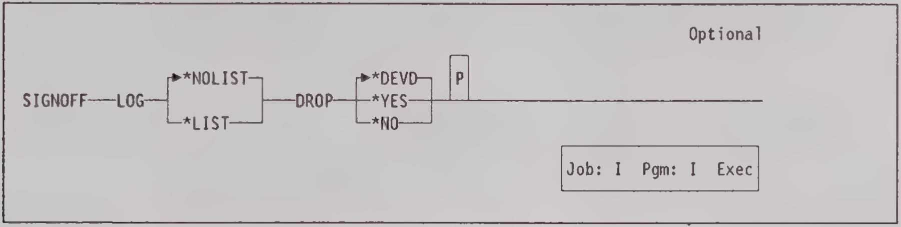
```
SETPGMINF  ROOTPGM(QTEMP/P1)  SUBPGM((QTEMP/P2) (QTEMP/P3))  +
  LIBFILE((*PASLIB))
```

This command links the application into a program to be processed.

SIGNOFF (Sign Off) Command

The Sign Off (SIGNOFF) command ends an interactive job or causes all jobs in a group to end. You enter this command to sign off at a work station or at the console.

Restriction: This command is valid only in an interactive job.



LOG Parameter
Specifies if the job log for this interactive job is deleted or is included in the job's spooled output for printing. This entry takes precedence over the LOG parameter value specified for the job.

***NOLIST:** The information in the job log, which has already been displayed throughout the job, is no longer needed and is to be deleted.

***LIST:** The job log is spooled for printing, along with the rest of the job's spooled output, if any.

DROP Parameter
Specifies, for switched lines only, if the switched line attached to the work station is disconnected (dropped) if no other work stations on the same line are signed on. This parameter is ignored if the work station is attached to a nonswitched line.

***DEVD:** The value specified in the DROP parameter of the work station's device description is assumed.

***YES:** The switched line is disconnected when the job is ended if no other work stations are signed on the line.

***NO:** The switched line is not disconnected when the job is ended.

Examples

```
SIGNOFF
```

This command signs off the user of the work station and ends the interactive job. The switched line is dropped only if so specified in the device description of this work station and if no other work station on this line is active. An end-of-job message that gives the job start and stop times is written in the job's log.

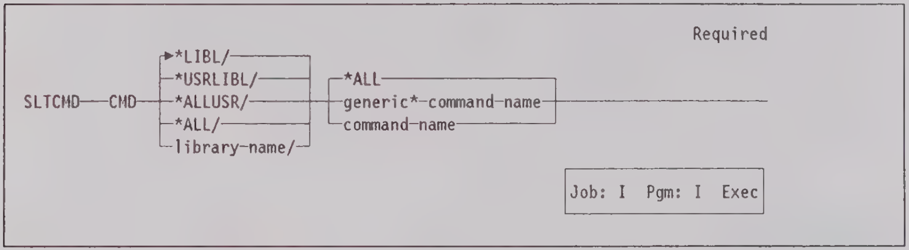
```
SIGNOFF  LOG(*LIST)  DROP(*NO)
```

This command ends the interactive job, but the switched line is not released. The job log is included with the job's spooled output.

SLTCMD (Select Command) Command

The Select Command (SLTCMD) command produces a list of commands from a user specified subset of command names. From this list, only the option to select the command is available.

Restriction: The user must have authority to both the command and the library where it is located for the operation he wants to perform.



CMD Parameter
Specifies the qualified name of the commands to be shown on the Select Command display.

The possible library values are:

<u>*LIBL</u>: All libraries in the user and system portions of the job's library list are searched.

***USRLIBL**: Only the libraries listed in the user portion of the library list are searched.

***ALLUSR**: All nonsystem libraries, including all user-defined libraries and the QGPL library, not just those in the job's library list, are searched. Libraries whose names begin with the letter Q, other than QGPL, are not included.

***ALL**: All libraries in the system, including QSYS, are searched.

library-name: Only the library named in this parameter is searched.

***ALL:** All commands (in the library(s) specified) are listed on the Select Command display.

generic-command-name:* Specify the generic name of the command being listed. A generic name can be specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified for the CMD parameter, then all commands that have names with the same prefix as the generic command name are listed.

For more information on the use of generic functions, refer to Appendix G, "Generic Functions."

command-name: Specify the name of the command being listed.

Example

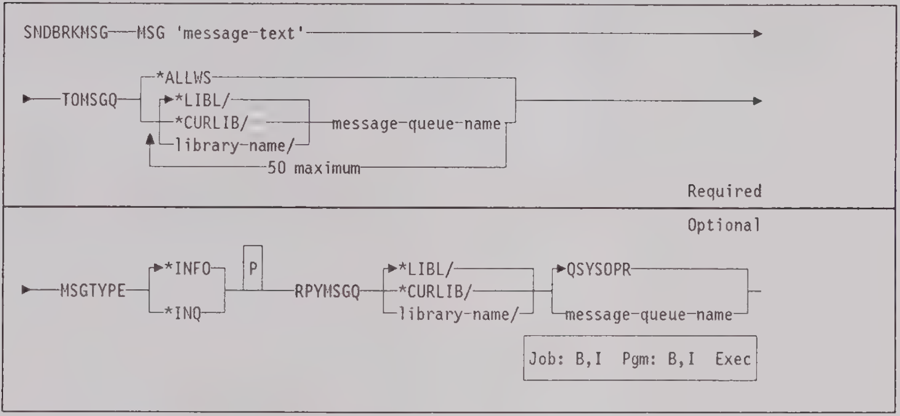
```
SLTCMD  CMD(QSYS/*ALL)
```

This command shows a list of all commands in library QSYS. The option to prompt and run commands is available.

SNDBRKMSG (Send Break Message) Command

The Send Break Message (SNDBRKMSG) command is used by the system operator to send an impromptu message to one or more work station message queues. (An impromptu message is a message that is not predefined and is not stored in a message file.) The command causes the message to be delivered always in break mode. The DSPMSG display is shown for the message when it is received, regardless of the setting of the message queue's delivery mode, severity, and break handling program. However, the message may not be displayed in some cases, depending on the BRKMSG job attribute. This command is primarily intended for the system operator's use.

Restriction: This command can be used to send break messages to work station message queues only.



MSG Parameter
Specifies the message text of the impromptu message being sent. The message must be enclosed in apostrophes if it contains blanks or other special characters. A maximum of 512 characters can be specified.

TOMSGQ Parameter
Specifies the qualified names of one or more work station message queues to which the break message is sent. Only the names of work station message queues can be specified.

***ALLWS:** The break message is sent to all work stations and Personal Computer message queues.

The possible library values are:

***LIBL:** The library list is used to locate the message queue.

***CURLIB:** The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message queue is located.

message-queue-name: Specify the name of the message queue to which the break message is being sent.

MSGTYPE Parameter
Specifies the type of message that is sent in break mode. Only informational or inquiry message types can be specified.

***INFO:** An information only message is sent in break mode.

***INQ:** An inquiry message is sent in break mode; the work station receiving the message is expected to reply to it. (Inquiry messages can be sent to only one work station at a time.)

RPYMSGQ Parameter
Specifies, only if an inquiry message is sent, the name of the message queue that the work station user's reply is sent to.

The possible library values are:

***LIBL:** The library list is used to locate the message queue.

***CURLIB:** The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message queue is located.

QSYSOPR: The replies to the break message are sent to the system operator's message queue, QSYSOPR.

message-queue-name: Specify the name of the message queue to which a reply to the break message is sent. Only a user or work station message queue can be specified.

Examples

```
SNDBRKMSG   MSG('The inventory application  +
  shuts down at 4:00 pm today.')
```

This command sends the message 'The inventory application shuts down at 4:00 pm today.' in break delivery mode to the message queues of all signed-on work stations. The message is delivered regardless of the delivery attribute setting of those message queues.

```
SNDBRKMSG   MSG('Your printed output is ready.')  +
  TOMSGQ(GEORGEMSGQ)
```

This example shows a typical use of the SNDBRKMSG command by the system operator to send a message to a work station user.

SNDDOC (Send Document) Command

The Send Document (SNDDOC) command calls the Send a Document display that allows you to send documents to one or more other users. If you use the SNDDOC command, you are not able to use the Attn key to interrupt the mail and calendar function.

Restrictions: You must be enrolled in the system distribution directory and in the mail and calendar function to use this command. You must also have AS/400 Office LPP installed.

```
SNDDOC——————————————————————————————————————————————————————————————

                                                ┌───────────────────────┐
                                                │Job: I  Pgm: I  Exec   │
                                                └───────────────────────┘
```

There are no parameters for this command.

Example

```
SNDDOC
```

This command displays the Send a Document display.

SNDDST (Send Distribution) Command

The Send Distribution (SNDDST) command allows a user to send distribution to a user or a list of users.

Restriction: If the current user of this command is working on behalf of another user, he must be given authority by the other user to work on his behalf using the Grant User Permission (GRTUSRPMN) command.

```
                                          Dependent
                                       Parameter Codes
                        ┌─*MSG──┐             A
SNDDST───TYPE───────────┼─*DOC──┼──────►      B
                        ├─*FILE─┤             C
                        └─*IDP──┘             D

►──TOUSRID──user-ID address or distribution-list-ID┐────────────────────►
   (ABCD)  ▲                                       │
           └──────────300 maximum──────────────────┘
                                           ┌─┐
                                           │P│
►──DSTD distribution-description───────────┘─┘──────────────────────────►
  (ABCD)
                                                                 Required
─────────────────────────────────────────────────────────────────────────
                                                                 Optional
         ┌►*NONE─────────┐
►──MSG───┤               ├──────────────────────────────────────────────►
  (ABCD) └─'message-text'┘

         ┌document-name─┐      ┌►*NONE──────┐
►──DOC───┤              ├──FLR─┤            ├───────────────────────────►
   (B)   └─*DOCID───────┘  (B) └─folder-name┘

          ┌►*NONE───────────────────────────────┐
►──DOCID──┤                                     ├───────────────────────►
   (B)    └─'library-assigned-document-name'────┘

            ┌►*LIBL/────────┐
►──DOCFILE──┼─*CURLIB/──────┼──data-base-file-name──────────────────────►
   (C)      └─library-name/─┘

           ┌►*FIRST──────┐            ┌►*CURRENT──────┐
►──DOCMBR──┤             ├──USRID─────┤               ├─────────────────►
   (C)     └─member-name─┘  (ABCD)    └─user-ID-address┘

            ┌►*DFT─────────────────┐          ┌►*DFT─────────┐
►──DOCTYPE──┼─*FFT─────────────────┼──SYSCOD──┤              ├──────────►
   (C)      ├─*RFT─────────────────┤   (C)    └─'system-code'┘
            └─document-type-number─┘
```

(continued on next page)

(continued from previous page)

```
             ┌►*NO──┐          ┌►*NORMAL─┐            ┌►*NO──┐
►──PERSONAL──┤      ├───PTY────┤         ├───CFMDEL───┤      ├──────────►
   (ABCD)    └─*YES─┘   (ABCD) └─*HIGH───┘   (ABCD)   └─*YES─┘

             ┌►*NONE─────────────────────────────────────────┐
►──IDPFILE───┼─*DOCFILE──────────────────────────────────────┼──────────►
   (CD)      │ ┌►*LIBL/─────────┐                             │
             └─┼─*CURLIB/───────┼───data-base-file-name───────┘
               └─library-name/──┘

            ┌►*FIRST──────┐          ┌►*DFT─────────────────────┐
►──IDPMBR───┤             ├───DOCD───┼─*DSTD────────────────────┼──────►
   (CD)     └─member-name─┘   (CD)   └─'document-description'───┘

            ┌►*NONE──────────────┐              ┌►*NONE──────────────┐
►──AUTHOR───┼─*USRID─────────────┼───DOCCLS─────┤                    ├──►
   (CD)     └─'document-author'─┬┘   (CD)       └─'document-class'───┘
             ▲                  │
             └──50 maximum──────┘

         ┌►*NONE───────────────┐               ┌►*NONE───────────────┐
►──KWD───┤                     ├───SUBJECT─────┼─*DOCD───────────────┼──►
   (CD)  └─'document-keyword'─┬┘   (CD)        └─'document-subject'─┬┘
          ▲                   │                 ▲                   │
          └──50 maximum───────┘                 └──50 maximum───────┘

             ┌►*NONE──────────┐     (1)      ┌►*NONE──────────────────────────┐
►──DOCDATE───┼─*CURRENT───────┼─────FILCAB───┤                                ├──►
   (CD)      └─document-date──┘     (D)      └─'filing-cabinet-reference'─────┘

            ┌►*NONE────────────┐              ┌►*NONE──────────┐
►──CPYLST───┤                  ├───EXPDATE────┤                ├────────►
   (CD)     └─'recipient-name'┬┘   (CD)       └─expiration-date┘
             ▲                │
             └──50 maximum────┘

             ┌►*NONE─────────┐               ┌►*NONE───────┐
►──CRTDATE───┼─*CURRENT──────┼───REFERENCE───┤             ├────────────►
   (CD)      └─create-date───┘   (CD)        └─'reference'─┘

             ┌►*NONE─────┐          ┌►*NONE──────────────┐
►──ACTDATE───┼─*CURRENT──┼──STATUS──┤                    ├───────────────►
   (CD)      └─date-due──┘  (CD)    └─status-of-document─┘

             ┌►*NONE──────────┐              ┌►*NONE─────┐
►──CMPDATE───┼─*CURRENT───────┼───PROJECT────┤           ├──────────────►
   (CD)      └─date-complete──┘   (CD)       └─'project'─┘

               ┌►*SYSVAL────────────────────────────┐
►──DOCCHRID────┼─*DEVD──────────────────────────────┼────────────────────►
   (C)         └─graphic-character-set  code-page───┘

       (2)     ┌►*SYSVAL────────────────────────────┐
►──CMDCHRID────┼─*DEVD──────────────────────────────┼─────────────────────
   (ABCD)      └─graphic-character-set  code-page───┘

(1) FILCAB is required when TYPE(*IDP) and IDPFILE(*NONE) are specified.
(2) This parameter applies to the following parameters:  TOUSRID, USRID,
    SYSCOD, MSG, DOCD, DSTD, AUTHOR, DOCCLS, KWD, FILCAB, CPYLST, SUBJECT,
    REFERENCE, STATUS, and PROJECT.

                                                ┌─────────────────────────┐
                                                │Job: B,I  Pgm: B,I  Exec │
                                                └─────────────────────────┘
```

TYPE Parameter

Specifies the type of information that is sent and the keywords that are valid on this command.

Dependent parameter codes for this parameter (see syntax diagram) determine which keywords are valid. If the dependent parameter code is listed in the syntax diagram, the keyword is valid when that type is specified.

***MSG:** Only the message specified in the MSG parameter is sent.

***DOC:** The document specified in the DOC or DOCID parameter is sent. The user must have authority for the document before it can be sent.

***FILE:** The data base file specified in the DOCFILE and the DOCMBR parameter is sent. The data base file is sent without any changes. The user must have authority for the data base file before he can send it.

***IDP:** The interchange document profile (IDP) specified in the IDPFILE and the IDPMBR parameters, or the document profile built from the keywords DOCD, AUTHOR, DOCCLS, KWD, SUBJECT, REFERENCE, STATUS, PROJECT, ACTDATE, CMPDATE, CRTDATE, DOCDATE, FILCAB, CPYLST, and/or EXPDATE. The profile can be a profile of a printed document.

TOUSRID Parameter

Specifies the user ID and address of one or more users to whom the distribution is being sent, or the distribution list name of one or more distribution lists containing the user ID and address of one or more users to whom the distribution is being sent. A combination of user IDs and distribution lists can be used on the same command.

DSTD Parameter

Specifies the description of the distribution. A maximum of 44 characters can be specified.

MSG Parameter

Specifies a short message.

***NONE:** No message is sent with the distribution.

'message-text': Specify the message to send to the users. A maximum of 256 characters can be specified.

DOC Parameter

Specifies the name of the document being sent.

document-name: Specify the user-assigned name of the document being sent. A maximum of 12 characters can be specified.

***DOCID:** The document being sent is identified by the library-assigned document name.

FLR Parameter

Specifies the name of the folder that contains the document being sent. This is the user-assigned name given to the folder when it is created. If the DOC parameter is specified, FLR parameter must also be specified.

***NONE:** No folder is specified when the document is identified by the DOCID parameter.

folder-name: Specify the name of the folder that contains the document being sent. A folder name can consist of a series of folder names (FLRA/FLRB/etc.) if the document being sent is located in a folder contained in another folder. A maximum of 63 characters can be specified.

DOCID Parameter

Specifies the library-assigned name of the document being sent. This is the name assigned to the document by the system when it was created. The library-assigned document names can be determined by using the Query Document Library (QRYDOCLIB) command or the library-assigned document name is returned in a completion code when using the File Document (FILDOC) command. Library-assigned document names are 24 characters in length with the following format: YYYYMMDDHHMNSSHSSNSNSNSN where:

YYYY = year

MM = month

DD = day

HH = hour

MN = minute

SS = second

HS = hundreth of a second

SNSNSNSN = system name

***NONE:** No library-assigned document name is required when the document is identified by the DOC parameter.

'library-assigned-document-name': Specify the library-assigned document name being sent.

DOCFILE Parameter

Specifies the name of the data base file that contains the document data to send. The data base file is a user-defined file, or the OUTFILE specified in the Receive Distribution (RCVDST), or the Retrieve Document (RTVDOC) commands. If an outfile is specified, only the document data record format is read from the outfile and the prefix is removed from the document data. More information on defining the format of data base files (OUTFILES) is in the *Data Base Guide*.

The possible library values are:

***LIBL:** The library list is used to locate the data base file.

***CURLIB:** The current library for the job is used to locate the data base file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the data base file is located.

DOCMBR Parameter

Specifies the document data base file member to send.

***FIRST:** The first member (in creation order) in the data base file is sent.

member-name: Specify the name of data base file to send.

USRID Parameter

Specifies the user ID and address that is associated with this request.

***CURRENT:** Distribution is sent for the current user of this command.

user-id/address: Specify the user ID and address of the user whose distribution is sent. The current user of this command must have the authority to work on behalf of the specified user profile. Authority to work on behalf of another user is given with the Grant User Permission (GRTUSRPMN) command.

DOCTYPE Parameter

Specifies the type of document being distributed. This identifier is used by the system to determine if it can handle the data stream correctly.

***DFT:** The system creates the right document type identifier is based upon where the data comes from.

Document type	*Description*
2	Final form text document
3	5520 revisable form text document
4	Word processing EBCDIC
5	Word processing information file
6	Image-data subset document
7	3730 text data stream
8	DIA document library descriptor document
9	3732 display document data stream
10	DIA display document data stream
11	Revisable form text document
12	1403 compatible data stream with variable length, unblocked records
13	Digitized ADS audio
14	IBM PC data file
15	Hard copy of document
223	Data file

If TYPE(*IDP) is specified, the type is 15 (printed document).

If TYPE(*FILE) is specified and the file is an OUTFILE created by the RTVDOC command, then the document type is taken from this file. If this file does not contain a document type or the value in the file for the document type is 0, then an error message is sent.

If the file is a user-defined file, then the document type is 223 (data file).

If TYPE(*DOC) is specified, the document type of the document sent is used.

***FFT:** The document is a final-form document. This type of document is intended to be shown and printed and not edited by the receiver.

***RFT:** The document is a revisable-form text. This type of document can be viewed, printed and edited by the receiver.

document-type-number: Specify a value from 2 through 65,535. The numbers from 2 through 32,767 are controlled by registering them with the IBM Document Interchange Architecture and are used for IBM defined document types. The numbers 32,768 through 65,535 are not registered with IBM and can be used for non-IBM defined document types. The meaning of these document types must be determined with the value of the SYSCOD parameter.

SYSCOD Parameter

Specifies the text that is used with the document type parameter to help uniquely identify the type of document being sent. The receiver of the data stream determines the document data stream and processing requirements to edit, view, print or change the document.

***DFT:** The system supplies a default system code. If the DOCTYPE value is a number from 1 through 32,767, the default is 'IBM AS/400 CL'. If the DOCTYPE is from 32,768 through 65,535, a system code must be specified.

'system-code': Specify the text that helps uniquely identify the type of document being sent. A maximum of 13 characters can be specified.

PERSONAL Parameter

Specifies if the distribution is sent as a personal distribution. A user working on behalf of another user cannot receive this distribution.

***NO:** The distribution is not sent as personal.

***YES:** The distribution is sent as personal.

PTY Parameter

Specifies if the distribution is sent using normal or high priority. For distributions to remote receivers, the priority determines which 'SNADS Next System Queue' is used. Normal priority distributions use the normal next system queue. High priority uses the priority next system queue. The difference between high and normal priority depend on information, specified by the user, on the Configuration Distribution Services (CFGDSTSRV) command. The handling of priority distributions by other office system nodes may vary, but generally the high priority distributions take the faster path when there is a choice of paths. For distribution to local receivers, the priority determines if a message is sent to the receiver's message queue to notify the receiver of the distribution. No message is sent for normal distributions.

***NORMAL:** Normal priority is used.

***HIGH:** High priority is used.

CFMDEL Parameter

Specifies if the sender wants a confirmation of delivery notification when each receiver receives the distribution. If confirmation of delivery is not requested, the sender is still informed if the distribution is not delivered because of system failures, routing failures, or has an invalid user ID. If confirmation of delivery is requested, the sender is informed when the receiver receives the distribution, or if the receiver deletes the distribution before receiving it. To get this information, the sender must do a Query Distribution (QRYDST) command with the OPTION(*OUT) specified.

***NO:** No request for confirmation of delivery is specified.

***YES:** Request a confirmation of delivery.

IDPFILE Parameter

Specifies the name of a data base file that contains the interchange document profile (IDP) data. The interchange document profile is a user-defined file or the OUTFILE specified on the Receive Distribution or Retrieve Distribution commands. If an OUTFILE is specified, only the data record format is read from the OUTFILE and the prefix is removed from the IDP data.

If this parameter is specified, the remaining parameters on this command are ignored except for the DOCHRID and CMDCHRID parameters.

***NONE:** The interchange document profile is not in a data base file. The information is provided by other keywords on this command.

***DOCFILE:** The interchange document profile is in the same file as the document. If a member is specified in the IDPMBR parameter, it is ignored.

The possible library values are:

***LIBL:** The library list is used to locate the data base file.

***CURLIB:** The current library for the job is used to locate the data base file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the data base file is located.

data-base-file-name: Specify the name of the data base file (library-name/data-base-file-name) that contains the interchange document profile. More information on defining the format of data base files (OUTFILES) is in the *Data Base Guide*.

IDPMBR Parameter

Specifies the interchange document file member name to send.

***FIRST:** The first member (by creation date) in the data base file is sent.

member-name: Specify the name of the data base file member to send.

DOCD Parameter

Specifies a description for the document being distributed. This is the Document Interchange Architecture IDP document name field.

***DFT:** The system creates a document description for data base files, the default is (library-name/file-name/member-name). For a description of a printed document (TYPE(*IDP)), the default is the distribution description.

***DSTD:** The distribution description is used for the document name.

'document-description': Specify the description of the document. A maximum of 44 characters can be used.

AUTHOR Parameter

Specifies the authors of the document.

***NONE:** No author is identified for the document.

***USRID:** The user ID address of the user who is requesting the distribution is used as the author's name.

'document-author': Specify the name of the authors. A maximum of 50 authors can be specified.

DOCCLS Parameter

Specifies the class associated with this document, such as MEMO, FORM, or SHEET.

***NONE:** No class is assigned to the document.

'document-class': Specify the document class. A maximum of 16 characters can be specified.

KWD Parameter

Specifies the keywords that can be used to describe the document.

***NONE:** No keywords are defined for this document,

'document-keyword': Specify the keywords to use to describe the document. A maximum of 50 keywords can be specified and each keyword can have a maximum of 35 characters.

SUBJECT Parameter

Specifies the subjects of the document.

***NONE:** No subjects are defined for the document.

***DOCD:** The document description is used as the subject for the document.

'document-subject': Specifies the subjects of the document. A maximum of 50 subjects can be specified and each subject can have a maximum of 60 characters of text.

DOCDATE Parameter

Specifies any date the user wants to assign to the document.

***NONE:** No date is assigned to the document.

***CURRENT:** The system assigns the current system date to the document.

document-date: Specify the value used as the document date. The date must be in the format specified by the system value QDATFMT.

FILCAB Parameter

Specifies the location of the document. This parameter is intended for printed documents. The Interchange Document Profile that refers to the printed document is distributed. This parameter is required if TYPE(*IDP) and IDPFILE(*NONE) are specified.

***NONE:** No filing cabinet reference is defined for this document.

'filing-cabinet-reference': Specify the text that describes where the printed document is located. A maximum of 60 characters can be specified.

CPYLST Parameter

Specifies names of the users who receive this distribution.

***NONE:** No copy list is included for this distribution.

'recipient-name': Specify the names of the users to receive the document. A maximum of 50 names can be specified and each name can have a maximum of 60 characters.

EXPDATE Parameter

Specifies the date when the document is no longer needed.

***NONE:** An expiration date is not specified.

expiration-date: Specify the value to use as the expiration date. The date must be specified in the format specified by the system value QDATFMT.

CRTDATE Parameter

Specifies the date the document was created.

***NONE:** No create date is specified.

***CURRENT:** The current system date is used as the date the document was created.

create-date: Specify the value used as the create date. The date must be specified in the format specified by the system value QDATFMT.

REFERENCE Parameter

Specifies a reference associated with the document.

***NONE:** No reference field is included in the Interchange Document Profile.

'reference': Specify the text that describes the reference associated with the document. A maximum of 60 characters can be used.

ACTDATE Parameter

Specifies the date when the action requested is due.

***NONE:** No action due date is specified.

***CURRENT:** The current system date is used as the action due date.

due-date: Specify the value used as the action due date. The date must be specified in the format specified by the system value QDATFMT.

STATUS Parameter

Specifies the user-defined status of the document. Examples of status are: In Progress, Pending Approval, or Retired.

***NONE:** No status is included in the Interchange Document Profile.

status-of-document: Specify the text that describes the status of the document. A maximum of 20 characters can be specified.

CMPDATE Parameter

Specifies the date when the action requested is completed.

***NONE:** No completion date is included.

***CURRENT:** The current system date is used as the completion date.

date-complete: Specify the value used as the completion date. The date must be specified in the format specified by the system value QDATFMT.

PROJECT Parameter

Specifies the project that the document is associated with.

***NONE:** No project field is included in the Interchange Document Profile.

'project': Specify the text that describes the project of the document. A maximum of 10 characters can be specified.

DOCCHRID Parameter

Specifies the character identifier (graphic character set and code page) for the data being distributed. The character identifier is related to the display device that was used to create the data being distributed. (For more information about CHRID processing, refer to the *Data Base Guide*.)

***SYSVAL:** The system determines the graphic character set and code page values for the command parameter from QCHRID system value.

***DEVD:** The system determines the graphic character set and code page values for the data being distributed from the display device description where this command was entered. This option is valid only when specified from an interactive job. If specified in an interactive CL program or a batch job, an error occurs.

graphic-character-set code-page: Specify the graphic character set and values that were used to create the data being distributed.

CMDCHRID Parameter

Specifies the character identifier (graphic character set and code page) for the data being entered as command parameter values (such as the text for the SUBJECT parameter or the user IDs for the TOUSRID parameter). The character identifier is related to the display device that was used to enter the command. (More information about CHRID processing is in the *Data Base Guide*.)

The CMDCHRID parameter applies to the following parameters and means in some cases that the data is translated to a code page and character set that is interchangeable with other IBM office products. The interchangeable code page and character set is '256 337'. In other cases, the code page and character set are attached to the field and sent with the field to allow the receiving terminal to correctly print and display the field.

Parameter	Translate
TOUSRID	Translated
SYSCOD	Translated
MSG	Translated
DOCD	Passed
DSTD	Translated
AUTHOR	Passed
DOCCLS	Passed
KWD	Passed
SUBJECT	Passed
FILCAB	Passed
CPYLST	Passed
REFERENCE	Passed
STATUS	Passed
PROJECT	Passed

The value translates the USRID parameter to character set and code page of '930 256'. The &dsnadmnl. contains the character set and code page table for '930 256'.

<u>*SYSVAL:</u> The system determines the graphic character set and code page values for the command parameter from the QCHRID system value.

***DEVD:** The system determines the graphic character set and code page values for the command parameter from the display device description where this command was entered. This option is valid only when specified from an interactive job. If specified in an interactive CL program or a batch job, an error occurs.

graphic-character-set code-page: Specify the graphic character set and values that were used to create the command parameter.

Example

```
SNDDST  TYPE(*MSG) TOUSRID(USER1 RCHSYS1) +
   DSTD('message to USER1') +
   MSG('HELLO USER1 at ADDRESS RCHSYS1') +  -
   CFMDEL(*YES)
```

This command sends a message-only distribution to USER1 at RCHSYS1 with a confirmation of delivery requested.

SNDEMLIGC (Send Japanese 3270PC Emulation Code) Command

The Send Japanese 3270PC Emulation Code (SNDEMLIGC) command sends a list of instructions to the system to do downloading of 5250 Personal Computer code from an AS/400 system to a 5550 Personal Computer. Download is needed only when a new release of the system is applied or changes have been made to the 5550 Personal Computer version code.

The download code is used to do SNA Japanese 3270PC emulation using the 5550 Personal Computer. Emulation of SNA Japanese 3270PC display stations and printers connected to a systems network architecture (SNA) 3274 is done from the download code. SNA Japanese 3270PC emulation is called through the Start 3270 Display Emulation (STREML3270) command. The STREML3270 command must specify IGCEMLPC(*YES) requesting SNA Japanese 3270PC emulation. Additional information on SNA Japanese 3270PC emulation is contained in the *SNA Japanese 3270PC Emulation User's Guide*, SC18-2213.

```
   (1)
SNDEMLIGC──────────────────────────────────────────────────────────────

(1) DBCS systems only.

                                                   ┌──────────────────┐
                                                   │Job: B,I  Pgm: B,I│
                                                   └──────────────────┘
```

There are no parameters for this command.

Example

```
SNDEMLIGC
```

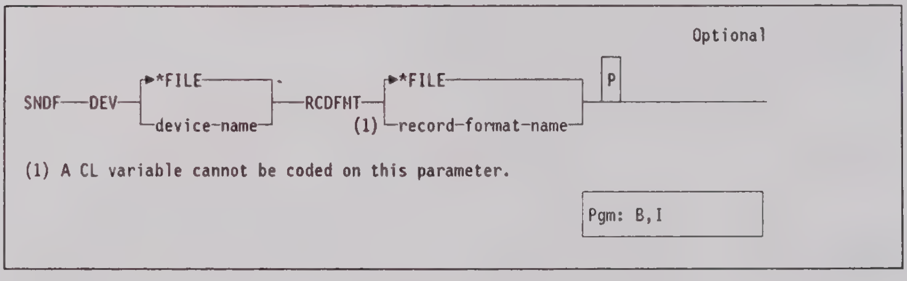
The SNDEMLIGC command allows PC commands to be run to send the code used to do SNA Japanese 3270PC emulation to the 5550 PC.

SNDF (Send File) Command

The Send File (SNDF) command is used by a CL program to send a record to a display device that is being used by an interactive user. The device can be any display station including the console. The command sends the data from the program's CL variables to the display's device file in the specified record format. These variables were automatically declared in the program (one for each field in the record format) when the CL source program was compiled and a Declare File (DCLF) command was processed as part of the source.

Of the record formats specified in the DCLF command, only one can be specified in each SNDF command. If the device file has not been opened, it is opened by this command. The file and record format specified in this command can be overridden by an Override with Display File (OVRDSPF) command if that command is entered before the file is opened. However, care should be taken that the fields in the overriding record format correspond to the CL variables declared in the program.

Restrictions: (1) This command is valid only within a CL program. (2) This command is valid only for display files. (3) It cannot be used with data base files.



(1) A CL variable cannot be coded on this parameter.

DEV Parameter

Specifies the name of the display device to which the data in the CL variables for the specified record format is sent.

***FILE:** The program's data is sent to the device associated with the device file (the device file that was declared in the FILE parameter of the DCLF command). If more than one device name is specified in the device file, *FILE cannot be specified.

device-name: Specify the name of the device or the name of the CL variable that contains the name of the device to which the program's data is sent.

RCDFMT Parameter

Specifies the name of the record format that is used to send data to the file. The format contains all the fields in the record. This parameter must be coded with a record format name if there is more than one record format name in the device file; *FILE cannot be coded if there is more than one. If the record format contains the INVITE DDS keyword (optioned on), the SNDF functions as if SNDRCVF WAIT(*NO) had been coded.

***FILE:** There is only one record format in the device file; that is the format in which the program's data is sent to the file.

record-format-name: Specify the name of the record format in which the program's data is sent to the file. A CL variable cannot be used here to specify the record format name.

Examples

```
DCLF   FILE(MENU1)
  •
  •
  •
SNDF
```

The record format in the device file MENU1 is sent to the device specified in the file. There is only one record format in the file.

```
DCLF   FILE(SCREEN1)  RCDFMT(REC1 REC2)
  •
  •
  •
SNDF   DEV(DISP3)  RCDFMT(REC1)
```

The device file named SCREEN1 causes the display station named DISP3 to display the data sent by the CL program to the user. The data is shown in the format specified by the REC1 record format.

```
DCLF   SCREEN1  (REC1 REC2)
  •
  •
  •
SNDF   *N  REC2
```

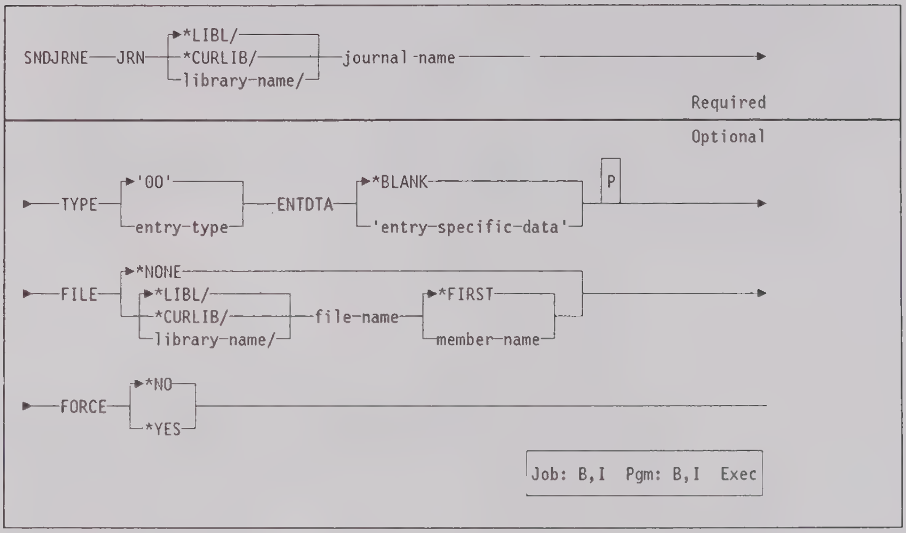
The device file named SCREEN1 is to send data to the device named in the same device file; *N is the null value for the DEV parameter coded positionally, which defaults to *FILE. The data is presented to the user in the format specified by REC2.

SNDJRNE (Send Journal Entry) Command

The Send Journal Entry (SNDJRNE) command is used to write a single journal entry to a specific journal. The entry can contain any information. The user may assign an entry type to the journal entry and may also associate the journal entry with a particular physical file member.

Note: The journal code for the entry is 'U', indicating a user-specified journal entry.

Restriction: If a file is specified, it must be currently journaled or must have been last journaled through the specified journal.



JRN Parameter

Specifies the qualified name of the journal that contains the new journal entry (library-name/journal-name).

The possible library values are:

***LIBL:** The library list is used to locate the journal.

***CURLIB:** The current library for the job is used to locate the journal. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the journal is located.

TYPE Parameter

Specifies the journal entry type of this journal entry.

00: The journal entry type is a '00' (hex F0F0).

entry-type: Specify a 2-character value or hex value used for the journal entry type. This value must be greater than or equal to hex C000.

If a hexadecimal value is specified that does not represent characters, that value is not shown on the DSPJRN display or the printout when the entry is displayed.

ENTDTA Parameter

Specifies the user-specified data that is placed in the variable portion of the journal entry.

***BLANK:** Specifies that no user-specified data is placed in the journal entry.

'entry-specific-data': Specify no more than 2000 characters, enclosed in apostrophes.

FILE Parameter

Specifies the qualified name of a data base physical file and member with which this entry is associated.

***NONE:** Specifies that there is no associated physical file for this entry.

The possible library values are:

***LIBL:** The library list is used to locate the file.

***CURLIB:** The current library for the job is used to locate the file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the file is located.

file-name: Specifies the name of the physical file with which this entry is associated (library-name/file-name).

***FIRST:** The entry is associated with the first member in the file.

member-name: Specifies the name of the physical file member with which this entry is associated.

FORCE Parameter
Specifies if the journal receiver is to be forced to auxiliary storage after the user entry is written to it.

***NO:** The journal receiver is not forced to auxiliary storage.

***YES:** The journal receiver is forced to auxiliary storage.

Examples

```
SNDJRNE   JRN(JRNLA)  TYPE(AB)  +
  ENTDTA('PROGRAM COMPLETE')  +
  FILE(MYLIB/ORDERENT MBR1)  FORCE(*YES)
```

This command places a journal entry of type AB (hex C1C2) with the user-created journal entry data on the current journal receivers attached to journal JRNLA as found using the library search list. The entry is associated with member MBR1 of file ORDERENT in library MYLIB. The journal receivers are forced to auxiliary storage after the entry has been placed on them.

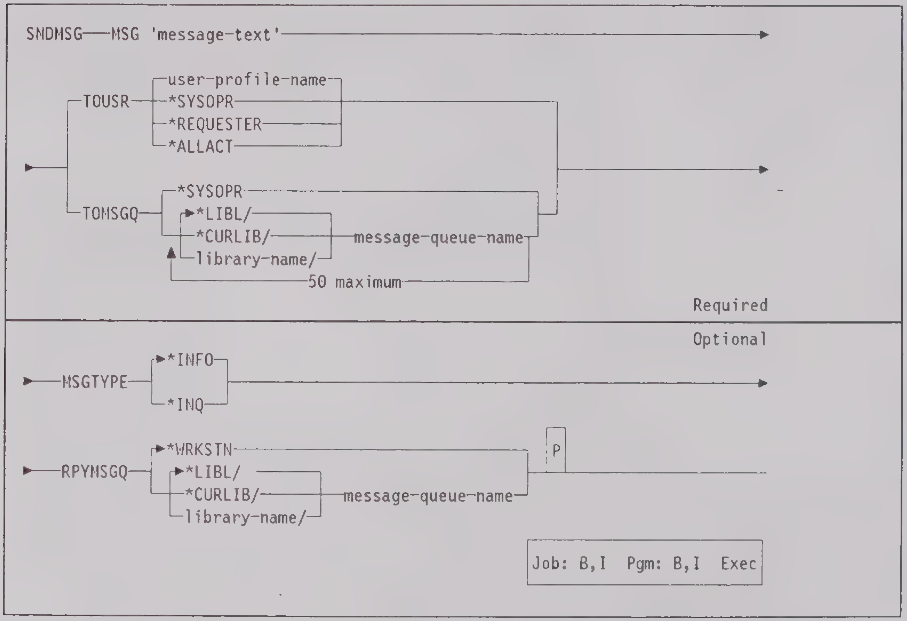
```
SNDJRNE   JRN(JRNLA)  TYPE(x'C1F1')
```

This command places a journal entry of type 'A1' (hex C1F1) with no user-created journal entry data on the current journal receivers attached to journal JRNLA as found using the library search list. The entry is not associated with any physical file member.

SNDMSG (Send Message) Command

The Send Message (SNDMSG) command is used by a display station user to send an impromptu message from his display station to one or more message queues. (An impromptu message is a message that is not predefined and is not stored in a message file.) The message can be sent to the system operator, to other display station users, to a user's message queue, or to the system history log, QHST. The sender can require a reply from the message receiver. The primary users of this command are display station operators and the system operator.

Restriction: To use this command, you must have *USE authority for the specified message queues and for the libraries in which they are stored. The SNDMSG command allows a message of up to 512 characters of first-level message to be sent and can be used to communicate between programs or to send long commands (requests) to a message queue.



MSG Parameter
Specifies the impromptu message that is being sent. The text must be enclosed in apostrophes if it contains blanks or other special characters. A maximum of 512 characters can be specified.

TOUSR Parameter
Specifies that the message is to be sent to the message queue specified in the user profile for the user named on this parameter. This parameter cannot be used if TOMSGQ is specified.

user-profile-name: Specify the user profile name of the user to whom the message is sent.

***SYSOPR:** The message is sent to the system operator.

***REQUESTER:** The message is sent to the user profile message queue for interactive jobs or to the system operator message queue (QSYS/QSYSOPR) for batch jobs.

***ALLACT:** A copy of the message is sent to the user profile message queue of each user profile with an interactive job currently running.

TOMSGQ Parameter
Specifies the qualified names of one to fifty message queues to which the message is being sent. A message may also be sent to one of the IBM-supplied logs (such as QHST). This parameter cannot be used if TOUSR is specified.

***SYSOPR:** The message is sent to the system operator message queue, QSYS/QSYSOPR.

The possible library values are:

***LIBL:** The library list is used to locate the message queue.

***CURLIB:** The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message queue is located.

message queue-name: Specify the qualified name of the message queue to which the message is being sent.

MSGTYPE Parameter

Specifies the type of message being sent. Only informational and inquiry message types can be specified.

***INFO:** An informational message is being sent.

***INQ:** An inquiry message is being sent; the message queue receiving the message must reply to it. If the receiver of a message does not respond to a message, the system automatically responds by sending the default reply. Inquiry messages are sent to only one message queue at a time.

RPYMSGQ Parameter

Specifies, only if an inquiry message is sent, the qualified name of the message queue to which a reply is sent.

***WRKSTN:** The reply to the message is sent to the display station message queue associated with the sender's display station.

The possible library values are:

***LIBL:** The library list is used to locate the message queue.

***CURLIB:** The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message queue is located.

message-queue-name: Specify the qualified name of the message queue to which a reply is sent. Only a user message queue, a display station message queue, or the system operator message queue can be specified.

Examples

```
SNDMSG   MSG('Do you want to update INV now?')  +
  TOUSR(JONES)  MSGTYPE(*INQ)  RPYMSGQ(SMITH)
```

This command sends a message to the user message queue JONES, requiring that a reply be sent to user message queue SMITH.

```
SNDMSG   MSG('Input errors on PAYROLL cost  +
  me 1 hour of run time.')  TOMSGQ(QHST)
```

This command is used by the system operator to send an informational message to the system's history log, QHST, through the log's message queue, which has the same name.

```
SNDMSG   MSG('Please make 2 copies of printer  +
  file LABORSTAT.')  TOMSGQ(QSYSOPR)
```

This example shows a typical use of the SNDMSG command by a display station user. The user is sending the message 'please make 2 copies of printer file LABORSTAT.' to the system operator.

```
SNDMSG   MSG ('How long will the display stations be  +
  online today?')  TOMSGQ(*SYSOPR)  MSGTYPE(*INQ)
```

This inquiry message from a display station user requires a reply. The system operator displays the message by using the DSPMSG command and enters the reply on the display. The display station user enters another DSPMSG command to display the reply.

SNDNETF (Send Network File) Command

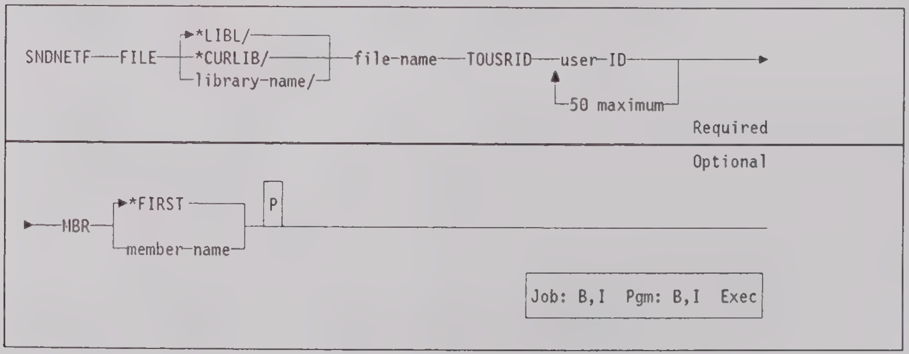
The Send Network File (SNDNETF) command sends a save file or a member of a physical data base file to another system user through the SNADS network. This command can be used to:

- Send data files to a user.
- Send source files to a user. Source sequence information is retained in the file sent.
- Send other object types stored in a save file to a user.

When the file arrives at its destination, a notification message is sent to both the recipient and sender of the file.

When a source physical file is sent, the source sequence number and change date in positions 1 through 12 of the record are sent with the file. These are retained if the file is received into a source physical file, and are truncated if the file is received into a nonsource physical file. When a file that was originally a nonsource physical file is received into a source physical file, the source sequence numbers are generated and placed in front of the records.

Restrictions: (1) You must be enrolled in the system distribution directory. (For a description of the system distribution directory, refer to the *CL Programmer's Guide.*) (2) The file that is sent must not contain more than approximately 16 million bytes of data. (3) The internal value for a node identifier may differ from the characters shown by the SNDNETF command depending on the type of work station (language) you are using. If the byte-string value specified for the TOUSRID command parameter does not match the rules for an internal node identifier value or if it does not match the internal value for any defined node (ignoring case differences), an error may be reported.



FILE Parameter

Specifies the name of the file that is sent. The file being sent can be a physical file or a save file; logical files and device files are not allowed. Overrides to the specified file are ignored.

The possible library values are:

***LIBL:** The library list is used to locate the file name.

***CURLIB:** The current library for the job is used to locate the file name. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the file name is located.

TOUSRID Parameter

Specifies the two-part user ID of one or more users to whom the file is being sent, or the name of one or more distribution lists containing the user IDs of one or more users to whom the file is being sent. A combination of user IDs and distribution lists can be specified on the same command. Each user ID or distribution list name is specified as a two-part name, and both parts are required.

Note: Depending on the type of work station you are using, the internal value for a user identifier may differ from the characters shown by the Display Directory (DSPDIR) command. If the byte-string value specified for the TOUSRID parameter does not match the rules for an internal user identifier value, or if it does not match the internal value for any enrolled user, an error may be reported.

MBR Parameter
Specifies the member that is sent from the file. A member name is not allowed if the file is a save file.

FIRST: The first member (in order of creation) in the file is sent.

member-name: Specify the name of the file member that is sent.

Example

```
SNDNETF  TOUSRID((TOM JONES))  FILE(EMPLOYEE)  MBR(PGMR)
```

This command sends member PGMR of file EMPLOYEE to the user identified to the network with a user ID of (TOM JONES). The library list is used to locate the file.

SNDNETMSG (Send Network Message) Command

The Send Network Message (SNDNETMSG) command sends a message to another system user through the SNADS network. This message is sent as an informational message to the message queue that is defined for the recipient on the receiving system. (For a detailed description of the SNADS network, refer to the *Communications Programmer's Guide.*)

Note: The recipient must have a valid message queue specified in his user profile. Messages sent with the SNDNETMSG command is rejected if the recipient does not have a message queue specified in his user profile. The message queue specified in the network attributes will not be used. For additional information on specifying a message queue when sending and receiving messages, see the *CL Programmer's Guide.*

Restrictions: (1) You must be enrolled in the system distribution directory. (For a description of the system distribution directory, refer to the *CL Programmer's Guide.*) (2) The internal value for a node identifier may differ from the characters shown by the SNDNETMSG command depending on the type of work station (language) you are using. If the byte-string value specified for the TOUSRID command parameter does not match the rules for an internal node identifier value or if it does not match the internal value for any defined node (ignoring case differences), an error may be reported.

```
                                                                 Required
                                                     ┌─┐
SNDNETMSG──MSG message-text──TOUSRID──user-ID────────┤P├──────────────────
                                      ▲           │  └─┘
                                      └─50 maximum─┘
                                                   Job: B,I  Pgm: B,I  Exec
```

MSG Parameter
Specifies the message text of the impromptu message that is sent. (An impromptu message is a message that is not stored in a message file.) The text must be enclosed in apostrophes if it contains blanks or special characters. A maximum length of 256 characters can be specified.

TOUSRID Parameter
Specifies one or more user IDs to whom an impromptu message is sent, or the name of one or more distribution lists containing user IDs of users to whom the message is to be sent. A combination of both user IDs and distribution lists can be specified on the same command. Each user ID or distribution list is specified as a two-part name, and both parts are required.

Note: Depending on the type of work station you are using, the internal value for a user identifier may differ from the characters shown by the Display Directory (DSPDIR) command. If the byte-string value specified for the TOUSRID parameter does not match the rules for an internal user identifier value, or if it does not match the internal value for any enrolled user, an error may be reported.

Example

```
SNDNETMSG MSG('Please do not make any more  +
  changes to the accounts receivable  +
  files for the next hour.')  +
  TOUSRID((PGMR SMITH))
```

This command sends the message specified in the MSG parameter to the user identified to the network with a user ID of (PGMR SMITH).

SNDNETSPLF (Send Network Spool File) Command

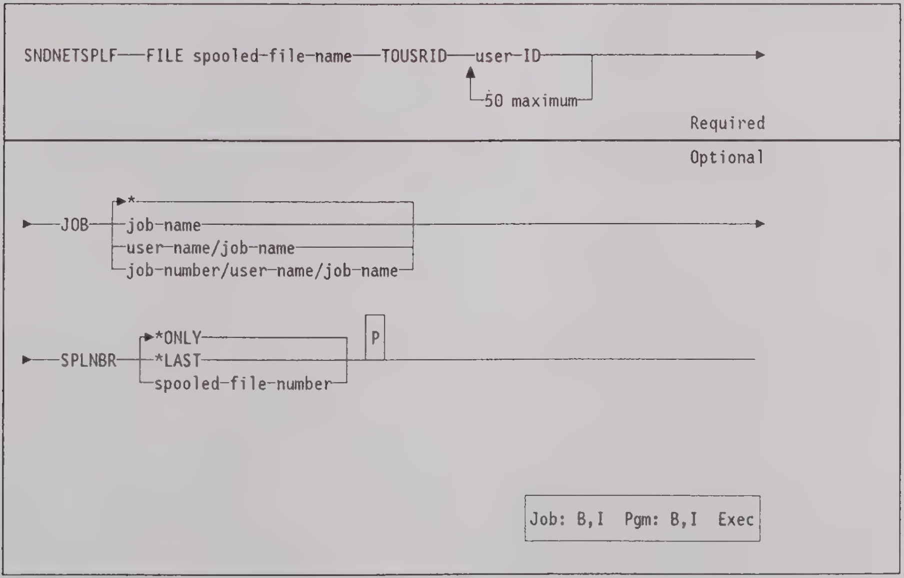
The Send Network Spool File (SNDNETSPLF) command sends a spooled file to another system user on the Systems Network Architecture Distribution Services (SNADS) network. The file is placed on an output queue that is specified in the user profile of the user to whom the spool file was sent. Spooled files that contain special device requirements cannot be sent using this command.

The following attributes of the spooled file are retained:

- File name
- Number of copies
- Characters per inch
- Drawer
- Form length
- Form type
- Form width
- Lines per inch
- Diskette label
- Diskette creation and end dates
- Diskette code type
- Diskette exchange type

When the file arrives at the destination system, a message is sent to both the recipient and sending user notifying them of the arrival of the spooled file.

Restrictions: (1) You must have read rights for the output queue containing the spooled file. (2) You must be enrolled in the system distribution directory to run this command. (3) The spooled file that is sent cannot contain more than approximately 16 million bytes of data. (4) The internal value for a node identifier may differ from the characters shown by the SNDNETSPLF command depending on the type of work station (language) you are using. If the byte-string value specified for the TOUSRID command parameter does not match the rules for an internal node identifier value or if it does differences), an error may be reported.



FILE Parameter

Specifies the name of the spooled file that is sent to the specified user. The file name is the name of the device file that was used by the program to produce the spooled output file.

The possible library values are:

***LIBL:** The library list is used to locate the file name.

***CURLIB:** The current library for the job is used to locate the file name. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the file name is located.

TOUSRID Parameter
Specifies the user ID of one or more users to whom the spooled file is sent, or the name of one or more distribution lists containing the user IDs of one or more users to whom the spooled file is to be sent. A combination of user IDs and distribution lists may be specified on the same command. Each user ID or distribution list name is specified as a two-part user name, and both parts are required. The user, or users, in the distribution list may be either remote or local.

Note: Depending on the type of work station you are using, the internal value for a user identifier may differ from the characters shown by the DSPDIR command. If the byte-string value specified for the TOUSRID parameter does not match the rules for an internal user identifier value, or if it does not match the internal value for any enrolled user, an error may be reported.

JOB Parameter
Specifies the name of the job that created the spooled output file whose data records are to be sent.

***:** The job that issued this command is the job that created the spooled file.

job-name: Specify the qualified name of the job that created the spooled file. If no job qualifier is given, all jobs currently in the system are searched for the simple job name. (For an expanded description of the JOB parameter, see Appendix A, "Expanded Parameter Descriptions.")

SPLNBR Parameter
Specifies the number of the spooled output file from the job whose data records are to be copied. (For an expanded description of the SPLNBR parameter, see Appendix A, "Expanded Parameter Descriptions.")

***ONLY:** Only one spooled file from the job has the specified name; therefore, the number of the spooled file is not necessary. If *ONLY is specified and more than one spooled output file has the specified name, an error message is issued.

***LAST:** The highest numbered spooled output file with the specified file name is sent.

spooled-file-number: Specify the number of the spooled file having the specified file name whose data records are copied.

Example

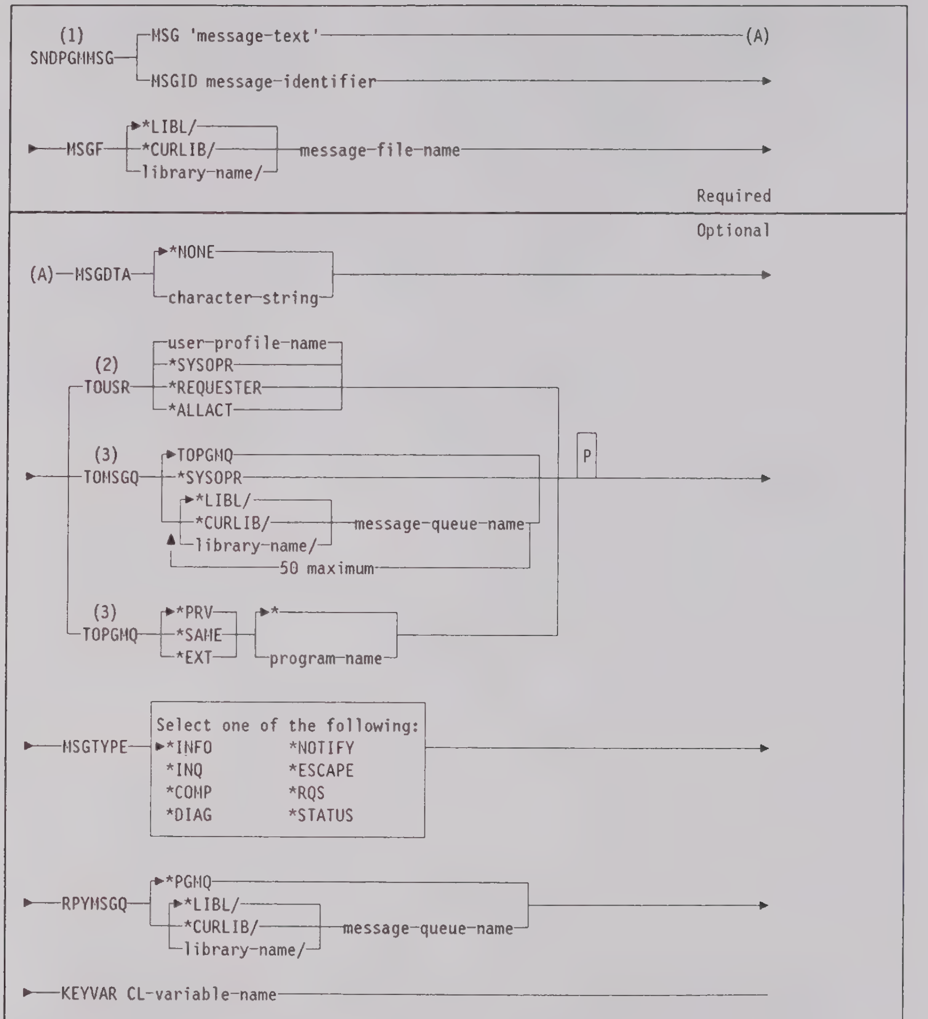
```
SNDNETSPLF  FILE(QPRINT)  TOUSRID((PRINTER PHOTO)) +
  JOB(PRINT.PAPER.142857)  SPLNBR(*LAST)
```

This command sends the last (most recently created) copy of spooled file QPRINT from job PRINT.PAPER.142857 to the user with a user ID of PRINTER PHOTO.

SNDPGMMSG (Send Program Message) Command

The Send Program Message (SNDPGMMSG) command is used by a program to send a program message to the named message queue, or to another program that uses the Receive Message (RCVMSG) command. The message can be an impromptu message or a predefined message stored in a message file. Substitution data can be specified in the MSGDTA parameter (as a single character string containing one or more concatenated message data fields) to replace the substitution variables in the message. The message can be sent to the job's external message queue, to the program message queue of a program called by the job, to one or more nonprogram message queues, such as a user profile message queue, the system operator message queue, or to the system history log, QHST.

Restrictions: (1) This command is valid only in CL programs. (2) The SNDPGMMSG command allows a message of up to 512 characters to be sent. However, only 76 characters are displayed on the Program Messages display. The Display Message (DSPMSG) command allows all 512 characters to be displayed.



(1) To code the following parameters **positionally**, you must code them in this order, using *N for those not being specified: MSG, MSGID, MSGF, and MSGDTA.
(2) TOUSR cannot be specified if TOMSGQ or TOPGMQ is specified.
(3) If both TOPGMQ and TOMSGQ are specified, the message is sent to the specified program message queue and to the specified nonprogram message queues.

Pgm: B,I

MSG Parameter

Specifies the message text of an impromptu message that is being sent by the program to a message queue. A maximum of 512 characters can be specified. The text string must be enclosed in apostrophes if special characters (including blanks) are used. If MSG is specified, MSGID cannot be specified. If MSG is specified, *ESCAPE, *NOTIFY, or *STATUS cannot be specified for the MSGTYPE parameter, because these types require that a message identifier must also be specified.

MSGID Parameter

Specifies the message identifier of a message description whose predefined message is being sent by the program to a message queue. If MSGID is specified, the MSG parameter cannot be specified.

MSGF Parameter

Specifies the qualified name of the message file that contains the predefined message being sent. If the message is sent to the QSYSOPR message queue, the library list used is the one that is in effect for the console or work station at which the message is actually displayed. This parameter is required if MSGID is specified.

The possible library values are:

***LIBL:** The library list is used to locate the message file.

***CURLIB:** The current library for the job is used to locate the message file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message file is located.

message-file-name: Specify the name of the message file which contains the predefined message being sent.

MSGDTA Parameter

Specifies the character string, or a CL variable that contains a character string, containing one or more substitution values that are used as message data fields within the predefined message. The substitution values take the place of the substitution variables that were defined in the message text when the message was defined.

Multiple values must be concatenated to form a single character string (see the *CL Programmer's Guide* for more details). If more than one substitution variable exists in the message, the specified character string must be the same length as the sum of the message data fields defined in the message. The length of the entire character string of concatenated substitution values cannot be greater than 512 characters. (For information about the length of individual message data fields, see the FMT parameter description in the ADDMSGD (Add Message Description) command.) Each substitution value, specified in the same order in the string as defined in the MSG and FMT parameters of the ADDMSGD command, must be specified as long as its associated variable was defined. Also, if the length of the message data that is passed to a substitution variable is shorter than the length specified for ADDMSGD FMT, the substitution value becomes a null field.

***NONE:** There are no program-supplied substitution values used in the specified message. The predefined message has no substitution variables in its text.

character-string: Specify the character string that gives the substitution values in the specified predefined message that is sent by the program, or specify the name of the CL variable that contains the character string.

If the message does contain more than one substitution variable, the character string containing the replacement values must be coded with the proper fill characters (zeros for numeric values, and blanks for character values) to position the values properly in the string to match the concatenated fields used to create the single string. If the string contains special characters (including blanks), it must be enclosed in apostrophes. The null value *N cannot be specified for a substitution value.

For example, if a predefined message contains the following three substitution variables:

Variable	Format Type	Length	Expected Values
&1	*CHAR	3 characters	Yes or no
&2	*CHAR	4 digits	4 digits
&3	*CHAR	8 characters	One or two words

the following valid character strings can be specified:

```
MSGDTA('Yes0007not done')
MSGDTA('No 0285finished')
MSGDTA(&DTA)
```

Note that the variable &2 is defined as a character variable, not as a decimal variable. If a character string is to be used instead of a CL variable to specify the substitution values in the MSGDTA parameter, the associated substitution variables must be defined (in the FMT parameter of the ADDMSGD command) as character variables to prevent invalid results in the text of the message sent.

The last example shows, instead of the actual character string, the CL variable &DTA, whose value is used as the message data. &DTA might have been declared in the sending program as:

```
DCL   VAR(&DTA)  TYPE(*CHAR)  +
  LEN(15)  VALUE('YES0175not done')
```

The character string, specified as the initial value for &DTA, contains the three concatenated substitution values `YES`, `0175`, and `not done`, which are to be sent as message data.

TOUSR Parameter

Specifies that the message is to be sent to the message queue specified in the user profile for the user named on this parameter. This parameter cannot be used if TOMSGQ or TOPGMQ is specified.

user-profile-name: Specify the user profile name of the user to whom the message is to be sent.

***SYSOPR:** The message is sent to the system operator message queue, QSYS/QSYSOPR.

***REQUESTER:** The message is sent to the user profile message queue for interactive jobs or to the system operator's message queue (QSYS/QSYSOPR) for batch jobs.

***ALLACT:** A copy of the message is sent to the user profile message queue of each user profile with an interactive job currently running.

TOMSGQ Parameter

Specifies the qualified names of one or more nonprogram message queues to which the message is sent by the program. This parameter cannot be used if TOUSR is specified.

***TOPGMQ:** The message is sent only to the program message queue specified on the TOPGMQ parameter.

***SYSOPR:** The message is sent to the system operator message queue, QSYSOPR.

The possible library values are:

***LIBL:** The library list is used to locate the message queue.

***CURLIB:** The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message queue is located.

message-queue-name: Specify the qualified names of the message queues to which the message is to be sent. (No more than fifty may be specified.)

TOPGMQ Parameter

Specifies the name of the program message queue to which the specified message is being sent. The named program message queue must be the queue of a program call in the sending job.

A list of two values is used to specify the program message queue to which the message is being sent. The first value specifies the relationship of the receiving program call with respect to the program call specified by the second value.

One of the following values can be specified for the first value:

***PRV:** The program message queue of the program called before the specified call (that is, the caller of the program) is the queue to which the message is sent.

***SAME:** The program message queue of the same program specified in the second value is the queue to which the message is sent.

***EXT:** The external message queue of the job is the queue to which the message is sent. The external message queue is used to communicate with the external requester of the job, such as a display station user. *INQ messages that are sent to *EXT will only wait 24 hours for a reply before the default reply is sent.

Messages sent to this queue can be 512 characters in length, but only 76 characters of text are shown on the PROGRAM MESSAGES display.

One of the following values can be specified for the second value:

***:** Specified as the second value in the list of values, it specifies the program message queue of the sending program.

program-name: Specified as the second value in the list of values, this value specifies the message queue of the most recent call of the named program. (The name of the program message queue is the same as the name of its corresponding program call.)

MSGTYPE Parameter

Specifies which message type is assigned to this message when it is sent by this program. Inquiry messages cannot be sent to program message queues. Completion, diagnostic, escape, notify, and status messages should be sent *only* to program message queues.

***INFO:** The message is sent as an informational message; it is to be sent to a program or nonprogram message queue.

***INQ:** The message is sent as an inquiry message; it is sent to a nonprogram message queue or to the external message queue of the job, *EXT. Normally, a reply must be returned to the program.

***COMP:** The message is sent as a completion message; it is sent to the program message queue of the program that requested work done by this program (sending the message). A completion message indicates the status of the work that is *successfully* performed. (Compare with **STATUS.*)

***DIAG:** The message is sent as a diagnostic message; it is sent to the program message queue of the program that requested work done by this program. Diagnostic messages provide information about errors detected by this program in the input sent to it or errors that occurred while it was running the requested function. Normally, an escape or notify message should be sent to inform the receiving program of the diagnostic messages that were sent to its program message queue.

***NOTIFY:** The message is sent as a notify message; it is sent to a program message queue. A notify message describes a condition for which the sending program requires corrective action or a reply from the program to which the message is sent before the sending program can continue processing the requested function. If control returns to the sending program, the reply can be received from its program message queue. *NOTIFY cannot be specified if the MSG parameter is specified.

***ESCAPE:** The message is sent as an escape message; it is sent to a program message queue. An escape message describes the error condition for which the sending program is being ended. Sending an escape message stops processing in the sending program. *ESCAPE cannot be specified if the MSG parameter is specified.

***RQS:** The message is sent as a request message; it is sent to the program message queue of the program that requested work done by this program.

A request message allows request data received from device files to be passed from this program to another. An impromptu message, specified by the MSG parameter, must be used to send the request.

***STATUS:** The message is sent as a status message; it is sent to a program message queue. A status message describes the status of work performed by the program sending the message. The first 28 characters of message data in the MSGDTA parameter are used as the comparison data for message monitors (established by the MONMSG command). If the program to which the status message is sent is not monitoring for that message identifier, control is returned to the sending program. If a status message is sent to the external message queue of an interactive job, the message is shown and processing continues; no response is required.

RPYMSGQ Parameter

Specifies, for inquiry and notify messages only, the qualified name of the program or nonprogram message queue to which the reply message is to be sent.

***PGMQ:** The reply to an inquiry or notify message (sent by this program) is sent to this program's own program message queue.

The possible library values are:

***LIBL:** The library list is used to locate the message queue.

***CURLIB:** The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message queue is located.

message-queue-name: Specify the qualified name of the message queue to which the reply is sent.

KEYVAR Parameter

Specifies the name of the CL character variable, if any, that contains the message reference key that identifies the message sent by the program containing this SNDPGMMSG command. The message reference key is assigned by the system when the message is sent and is placed in the variable specified by KEYVAR. The message key can then, for example, be used in the MSGKEY parameter of the RCVMSG command to receive the reply to an inquiry message being sent by this command. The message reference key can be used by the program specified on the TOPGMQ parameter for building message subfiles. The CL variable is the name of the field for which the SFLMSGKEY keyword is specified in the DDS for the message subfile.

If a message is being sent to a program message queue, KEYVAR refers to that message queue (specified by the TOPGMQ parameter). If *INQ or *NOTIFY is specified for the MSGTYPE parameter, KEYVAR refers to the message queue specified in the RPYMSGQ parameter. In all other cases, KEYVAR refers to the message queue specified in the TOMSGQ parameter.

If an inquiry message has been sent to the *EXT program message queue with a value specified for the KEYVAR parameter, the message reference key returned is that of the sender's copy message, which is located on the reply message queue. The reply message queue defaults to the program message queue of the program that sent the inquiry message.

Any type of message can be assigned a key when it is being sent to a program message queue. For messages sent to a nonprogram message queue, message reference keys are available for inquiry (*INQ) messages only. If another message type is sent to a nonprogram queue, no message key is available and blanks are returned for KEYVAR.

The variable must be a character variable having a length of 4 characters. If KEYVAR is not specified and a reply is required, it can be received by the program in FIFO order.

Examples

```
SNDPGMMSG   MSGID(UIN0023)  MSGF(INV)  +
  MSGDTA(' 50100')  TOPGMQ(*EXT)
```

This command sends the message identified as UIN0023, which is stored in message file INV, to the external message queue of the job. (The Program Message Display presents the message at a display station.) The data, containing two substitution values specified in the MSGDTA parameter is sent with the message. This data can then be used as substitution values when the message is received, or it can be used as data to be dumped, depending on how the message UIN0023 is defined in the message file. Assuming that the variables &1 and &2 have been defined in the message file as character variables, each 3 characters long, and that the message UIN0023 is:

'Requested item decreased by &1; current balance &2.'

the message text received is:

'Requested item decreased by 50; current balance 100.'

```
SNDPGMMSG   MSG('Mount payroll checks in printer  +
  before continuing')  MSGTYPE(*INQ)  +
  TOMSGQ(*SYSOPR)
```

This command sends an inquiry message to the system operator. The operator looks at the message using the DSPMSG command and responds to the message on that display. A Receive Message (RCVMSG) command is used in the program to accept the operator's response.

```
SNDPGMMSG   MSGID(USR0001)  MSGF(USRMSGR)  +
  TOPGMQ(*PRV  *)  MSGTYPE(*ESCAPE)
```

This command is an example of how a message could be sent to a program to cause an abnormal end. The message USR0001 could indicate that an invalid code was passed (such as a nonnumeric code when numeric is required). Because the message being sent is an escape message, the program sending the message is ended. The values *PRV and * did not have to be coded on this command because they are the default values on the TOPGMQ parameter.

SNDPTFORD (Send Program Temporary Fix Order) Command

The Send Program Temporary Fix Order (SNDPTFORD) command allows you to prepare an order for a specific fix or a fix package to be sent to your remote service support system.

You can prepare an order for the following:

- A corrective or preventive PTF package
- Summary information for available PTFs
- Preventive Service Planning (PSP) information

```
                                                        Required
                                  ┌─┐
                                  │P│
SNDPTFORD───PTFID PTF-number──────┴─┴───────────────────────────
                                              ┌──────────────────────┐
                                              │Job: B,I  Pgm: B,I Exec│
                                              └──────────────────────┘
```

PTFID Parameter

Specifies the ID for the PTF or PTF package that you are requesting.

PTF-number: Specify one of the following:

- PTF ID (7 characters)
- PTF summary information (SF97vrm), where v, r, and m are the version, release, and modification level for the PTF summary information.
- Preventative Service Planning (PSP) information (SF98vrm), where v, r, and m are the version, release, and modification level for the PSP package
- PTF package (SF99vrm), where v, r, and m are the version, release, and modification level for the PTF package.

Example

```
SNDPTFORD  PTFID(1234567)
```

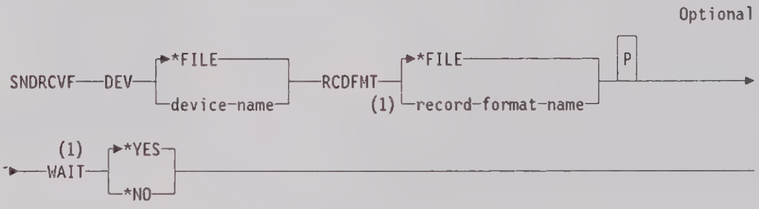
This command prepares a request for a fix with the number of 1234567.

SNDRCVF (Send/Receive File) Command

The Send/Receive File (SNDRCVF) command is used by a CL program to send to and receive data from a device that is being used interactively by a user. The data is passed between the program in which the SNDRCVF command is used and the display device identified in the command. The data is passed using the display device file that was declared in the program. (A Declare File (DCLF) command included in the source used to compile the program was used to declare the file.) The data for each send/receive operation is passed as one record in a format identified by the RCDFMT parameter of this command (the format is defined in DDS). One CL variable is used for each field of the record format to pass the data; the CL variables used (including DDS indicators) are declared implicitly.

Of the record formats specified in the DCLF command, only one can be specified in each SNDRCVF command. If the device file has not been opened, it is opened by this command. The file and record format specified in this command can be overridden by an Override with Display File (OVRDSPF) command if that command is entered before the file is opened. However, care should be taken that the fields in the overriding record format correspond to the CL variables declared in the program.

Restriction: This command is valid only within a CL program and only for display files. It cannot be used with data base files.



(1) A CL variable cannot be coded on this parameter.

Pgm: B,I

DEV Parameter
Specifies the name of the display device that the CL program's data is sent to and the user's data is received from. A CL variable can be specified on this parameter so that the device name can be changed without changing the command.

***FILE:** The data is sent to and received from the device associated with the device file (the device file that was declared in the FILE parameter of the DCLF command). If more than one device name is specified in the device file, *FILE cannot be specified.

device-name: Specify the name of the device or the name of the CL variable that contains the name of the device that the CL program sends data to and receives data from. If a CL variable name is used in this parameter, only one SNDRCVF command is needed in the program to receive data from several devices.

RCDFMT Parameter
Specifies the name of the record format that is used to pass the data between the CL program and the user. The format contains all the fields in the record. This parameter must be coded with a record format name if there is more than one record format in the device file; *FILE cannot be coded if there is more than one. SNDRCVF ignores the INVITE DDS keyword.

***FILE:** Only one record format is in the device file; that format is used to send the data to and receive the data from the user.

record-format-name: Specifies the name of the record format in which the data is sent to and received from the user. A CL variable name cannot be used here to specify the record format name.

WAIT Parameter
Specifies whether the CL program waits for the data to be received from the user's device or continue processing the commands that follow this SNDRCVF command. If WAIT(*NO) is specified, the program must issue a WAIT command later in the program to complete the input operation.

Note: A CL variable cannot be coded on this parameter.

<u>*YES:</u> The program waits until the input operation from the device is completed; the next command is not processed until then.

***NO:** The program does not wait for the input data; it continues to process commands until a WAIT command is reached later in the program.

Examples

```
DCLF   FILE(MENU1)
  •
  •
  •  -
SNDRCVF
```

This command sends and receives user data by way of the device file MENU1. Only one record format exists in the file. The device used is specified in the file.

```
DCLF   FILE(SCR)  RCDFMT(REC8)
  •
  •
  •
SNDRCVF   RCDFMT(REC8)
```

The CL program sends data to a user and receives data for the user who is using the device named in the device file (*FILE was assumed because DEV was not specified). The data is passed in the format specified by REC8 record format in the device file named SCR. The CL program waits for the user data before continuing.

```
DCLF   FILE(DF1)  RCDFMT(REC8)
  •
  •
  •
SNDRCVF   DEV(&DN)  RCDFMT(REC8)  WAIT(*NO)
  •
  •
  •
WAIT   DEV(&DN)
```

This command sends and receives user data by way of the device file named DF1. Using the record format REC8, the CL program can pass data between itself and the user who is at the device named in the variable &DN, but it does not wait for a response to come back. If the program is to send and receive data from several devices, the same SNDRCVF command can be used. Only the device specified by &DN for the DEV parameter needs to be changed. A WAIT command for each device must be issued later in the program to ensure that all the devices do respond.

SNDRPY (Send Reply) Command

The Send Reply (SNDRPY) command sends a reply message to the sender of an inquiry message. The message that is answered is the one having the specified message reference key that was received at the specified message queue. If the specified message queue is not allocated to the job in which this command is entered, it is implicitly allocated by this command for the duration of the command.

```
SNDRPY──MSGKEY CL-variable-name─────────────────────────────────►

       ┌►*LIBL/────────┐
►──MSGQ─┼─*CURLIB/──────┼─message-queue-name────────────────────►
       └─library-name/─┘
                                                        Required
─────────────────────────────────────────────────────────────────
                                                        Optional
         ┌►*DFT──────────┐ ┌─┐
►──RPY───┤               ├─┤P├──RMV──┬►*YES─┬──────────────────
         └─'reply-text'──┘ └─┘       └─*NO──┘

                                              ┌──────────┐
                                              │Pgm: B,I  │
                                              └──────────┘
```

MSGKEY Parameter
Specifies the name of the CL variable in the program that contains the message reference key of the message that the reply answers.

MSGQ Parameter
Specifies the qualified name of the message queue that received the inquiry message being answered.

The possible library values are:

***LIBL:** The library list is used to locate the message queue.

***CURLIB:** The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message queue is located.

RPY Parameter
Specifies the reply that the program is sends as a response to the inquiry message.

***DFT:** The default reply stored in the message description of the inquiry message that was sent is sent as the reply. (For impromptu inquiry messages, *DFT is the same as *N.)

'reply-text': Enter the text (enclosed in apostrophes if it contains blanks or special characters) or a CL variable that contains the text that is sent as the program's reply to the inquiry message. The number of characters and their format are defined by the validity specifications given in the Add Message Description (ADDMSGD) command for the specified inquiry message. However, if no validity specifications are specified for replies in the ADDMSGD command, as many as 132 characters can be used in the reply text.

RMV Parameter
Specifies whether the inquiry message and its reply are removed from the specified message queue.

***YES:** The message and its reply are removed from the message queue when the reply is sent.

***NO:** The message and its reply are retained in the message queue. The inquiry message cannot be replied to more than once, but it can be received or displayed multiple times.

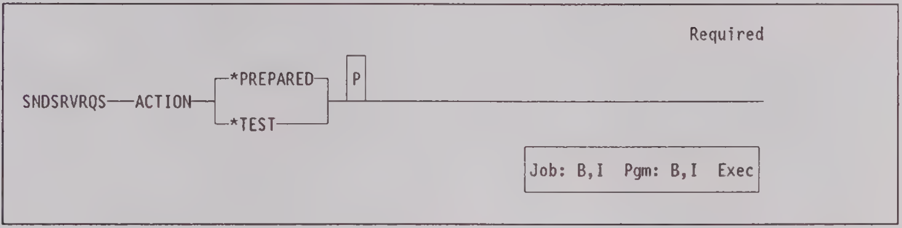
Example

```
SNDRPY   MSGKEY(&KEY)  MSGQ(SMITH)  RPY(YES)
```

This command sends a reply of YES to the message whose reference key is specified by &KEY, which was received at message queue SMITH. Because the reply contains only one word, the reply does not have to be enclosed in apostrophes.

SNDSRVRQS (Send Service Request) Command

The Send Service Request (SNDSRVRQS) command establishes a communications session and sends problem information to your service support system.



ACTION Parameter
Specifies the type of request(s) that is processed during this session.

***PREPARED:** All records in the problem log with a status of PREPARED are sent to the remote service support system.

***TEST:** A test is preformed on the remote support function and the communications link to it.

Example

```
SNDSRVRQS  ACTION(*PREPARED)
```

This command establishes the communications link to the service support system and sends all records in the problem log with the status prepared. The result of each problem log entry reported may be one of the following:

- PTFs sent to your system
- PTFs ordered from the code distribution center
- CE contacted automatically
- Service support center contacted automatically. The service support center representative will call you.
- Parts list

SNDTIEF (Send Technical Information Exchange File) Command

The Send Technical Information Exchange File (SNDTIEF) command sends a specified file to the IBM support systems.

```
                         ┌►*LIBL/────────┐        ┌►*FIRST──────┐
SNDTIEF──FILE file-name──┼─*CURLIB/──────┼──MBR───┤             ├──────────►
                         └─library-name/─┘        └─member-name─┘

                                                 ┌─┐
                                                 │P│
►──TOUSRID user-ID-name──TONODE system-name──────┴─┴─────────────────────────►
                                                                     Required
─────────────────────────────────────────────────────────────────────────────
                                                                     Optional
                             ┌►*OTHER──┐        ┌►2┐
►──TOFILE file-name──TYPE────┤         ├──PTY───┤  ├──────────────────────────
                             └─*SYSTPLD┘        └─1┘

                                                   Job: B  Pgm: B  Exec
```

FILE Parameter
Specifies the physical file to be sent to the IBM support center.

The possible library values are:

***LIBL:** The library list is used to locate the file.

***CURLIB:** The current library for the job is used to locate the file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the file is located.

MBR Parameter
Specifies the name of the member in the data base file that is to be transmitted to the IBM support system.

The possible members names are:

***FIRST:** The library list is used to locate the data base file.

member-name: Specify the library where the data base is located.

TOUSRID Parameter
Specifies the user who receives the file.

TONODE Parameter
Specifies the name of the system that receives the file.

TOFILE Parameter
Specifies the name that the file has on the receiving system.

TYPE Parameter
Specifies the type of contents of the file.

***OTHER:** The content of the file is not specified.

***SYSTPLD:** The file contains system configuration descriptions.

PTY Parameter
Specifies the transmission priority that is assigned to the file.

*2: Indicates normal priority.

*1: Indicates the highest priority.

Example

```
SNDTIEF  FILE(QSYS/MYTOPO)  TOUSRID(IBMSERV)  +
  TOSYS(IBMTIE)  TOFILE(ACMETOPO)
```

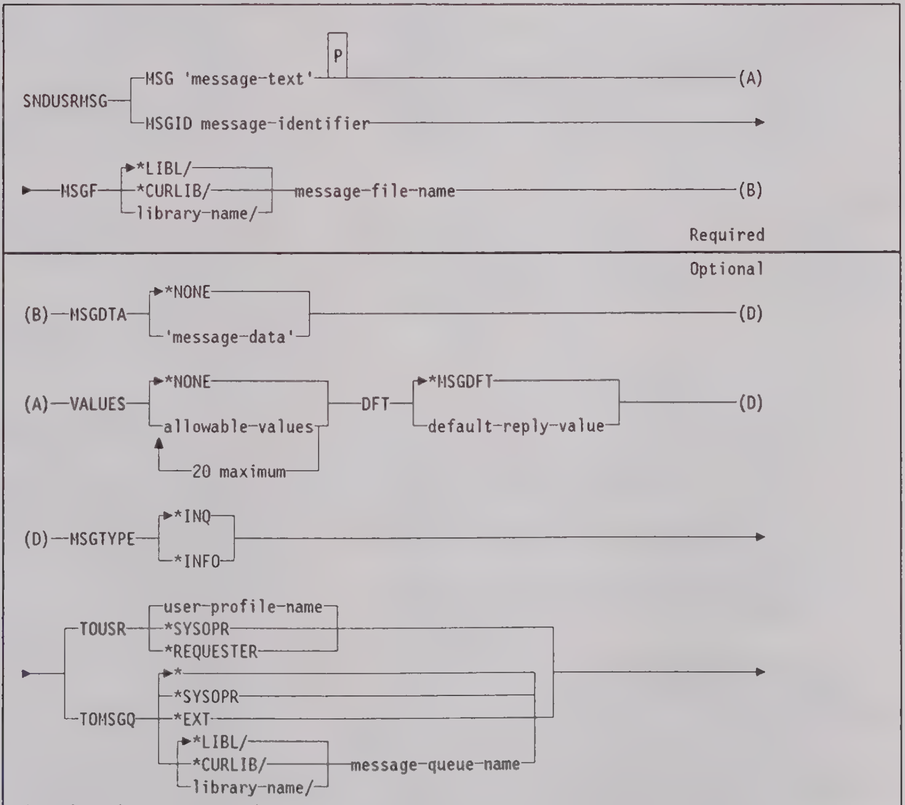
This command sends a file named MYTOPO from library QSYS to TIE. It is held in a mailbox for user IBMSERV on system IBMTIE. When it is received by the user, it will be named ACMETOPO.

SNDUSRMSG (Send User Message) Command

The Send User Message (SNDUSRMSG) command is used by a program to send a message to a message queue and optionally receive a reply to that message. The message sent using this command can be either an impromptu or a predefined message and can be sent to a display station user in an interactive job or a specific message queue. For inquiry messages, a CL variable can be specified to receive the reply value.

This command uses a combination of parameters available on the Send Program Message (SNDPGMMSG) and Receive Message (RCVMSG) commands to allow the sending and receiving of messages from a program with a single command. Also, the SNDUSRMSG command provides validity checking and uppercase translation for replies to inquiry messages.

Restrictions: (1) This command is valid only in CL programs. (2) The SNDUSRMSG command allows a message of up to 512 characters of first-level message to be sent. However, only 80 characters are displayed on the Program Messages display. (The Display Message (DSPMSG) command allows all 512 characters to be displayed.)



(continued on next page)

```
(continued from previous page)
                                    ┌►*NONE─────────────────────────────────┐
►──MSGRPY CL-variable-name──TRNTBL──┤ ┌►*LIBL/─────────┐  ┌►QSYSTRNTBL─┐    ├────
                                    └─┼─*CURLIB/───────┼──┤            ├────┘
                                      └─library-name/──┘  └─table-name─┘

                                                          Pgm: B,I
```

MSG Parameter

Specifies the message text of an impromptu message that is sent by the program. A maximum of 512 characters can be specified. The text string must be enclosed in apostrophes if special characters (including blanks) are used. If a value is specified for the MSG parameter, the MSGID, MSGF, and MSGDTA parameters cannot be specified.

MSGID Parameter

Specifies the message identifier of a predefined message sent by the program to a message queue. If a value is specified for the MSGID parameter, a value must also be specified for the MSGF parameter, while the MSG and VALUES parameters cannot be specified.

MSGF Parameter

Specifies the qualified name of the message file that contains the predefined message (specified in the MSGID parameter) being sent. The MSGF parameter is valid only if a predefined message is being sent.

The possible library values are:

*LIBL: The library list is used to locate the message file.

*CURLIB: The current library for the job is used to locate the message file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library in which the message file is located.

message-file-name: Specify the message file containing the predefined message.

MSGDTA Parameter

Specifies the character string or the CL variable that contains a character string used as the message data in the predefined message. A character string that is specified contains one or more substitution values that are used in place of the substitution variables that were defined in the message's text when the message was defined. The MSGDTA parameter is valid only if a predefined message is being sent. For more information, refer to the description of the MSGDTA parameter of the Send Program Message (SNDPGMMSG) command.

*NONE: No message data is specified for the predefined message.

'message-data': Enter the character string that gives the substitution values in the specified predefined message that is sent, or specify the name of the variable that contains the character string.

VALUES Parameter

Specifies a list of valid replies to an inquiry message sent by this command. No more than 20 replies can be specified in the list. If the reply to the inquiry message does not match one of the specified values, an error message is sent to the reply's sender and the inquiry message is sent again.

Translation of the reply, if requested (on the TRNTBL parameter), will occur before verification of a match between the reply and the specified value. If you use TRNTBL(*LIBL/QSYSTRNTBL) to translate the reply values from lowercase to uppercase and lowercase characters appear in a specified value, no match occurs.

Variables can be used in the list of replies. This parameter is valid only if MSGTYPE(*INQ) is specified.

*NONE: No replies to inquiry messages are specified. Any reply to an inquiry message is valid.

allowable-value: Specify no more than 20 values that are compared to replies received for inquiry messages sent by this command. The maximum length of each value is 32 characters.

DFT Parameter

Specifies the value used as the reply to an inquiry message (sent by this command) if the inquiry message is sent to a message queue that is in default delivery mode, or for any other reason the default reply is sent. The value specified here as the default reply does not have to match an entry in the list of valid replies specified in the VALUES parameter. The default reply value is not translated or validity-checked. This parameter is valid only if a value is specified for the MSGRPY parameter.

*MSGDFT: No default reply to inquiry messages is specified.

'default-reply-value': Enter the reply (enclosed in apostrophes) used as the default reply.

MSGTYPE Parameter

Specifies the type of message sent. Only inquiry and informational messages can be specified.

***INQ:** An inquiry message is sent; the message queue receiving the message must reply to it.

***INFO:** An informational message is sent. Translation of the reply, if requested (see TRNTBL parameter), will occur before the verification that the reply matches one of the values specified. If you use TRNTBL(*LIBL/QSYSTRNTBL) to translate lowercase to uppercase, and lowercase characters appear in any value specified, no match can occur with that value.

TOUSR Parameter

Specifies that the message is to be sent to the message queue specified in the user profile for the user named on this parameter. This parameter cannot be used if TOMSGQ is specified.

user-profile-name: Specify the user profile name of the user to whom the message is sent.

***SYSOPR:** The message is sent to the system operator message queue, QSYS/QSYSOPR.

***REQUESTER:** The message is sent to the user profile message queue for interactive jobs or to the system operator's message queue (QSYS/QSYSOPR) for batch jobs.

TOMSGQ Parameter

Specifies the qualified name of the message queue to which the message is sent. This parameter cannot be used if TOUSR is specified.

***:** In an interactive job, the message is sent to the external message queue (*EXT). In a batch job, the message is sent to the system operator (QSYSOPR) message queue.

***SYSOPR:** The message is sent to the system operator message queue, QSYSOPR.

***EXT:** The message is sent to the job's external message queue. For batch job inquiry messages, the default reply is always received. Messages sent to this queue can be 512 characters in length, but only 80 characters are displayed on the Program Messages display.

The possible library values are:

***LIBL:** The library list is used to locate the message queue.

***CURLIB:** The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message queue is located.

message-queue-name: Specify the name and library of the message queue that is to receive the message being sent.

MSGRPY Parameter

Specifies the CL character variable (of up to 132 characters) that contains the reply received in response to an inquiry message. This parameter is valid only if MSGTYPE(*INQ) is specified. If no value is coded for this parameter when MSGTYPE(*INQ) is specified, the command awaits a response from the operator in reply to the inquiry message. If the reply is longer than the variable, it is truncated; if it is shorter, it is padded with blanks.

TRNTBL Parameter

Specifies, if the reply value is being translated, the qualified name of the translate table used.

***NONE:** The reply is not being translated.

The possible library values are:

***LIBL:** The library list is used to locate the translate table.

***CURLIB:** The current library for the job is used to locate the translate table. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the translate table is located.

QSYSTRNTBL: The translate table named QSYSTRNTBL (found in library *LIBL) is used to translate the reply value.

The IBM-supplied translate table QSYSTRNTBL table translates, for the English language only, all lowercase characters in the range of X'81' to X'A9' to uppercase characters. All other characters are not translated.

If you want to use a different translate table, you can do so by using the Create Table (CRTTBL) command and by specifying that particular table for this parameter.

table-name: Specify the name of the translate table used to translate the message reply.

Examples

```
SNDUSRMSG   MSG('Data has been verified.  Do you +
  want to update the master files (Y or N)?') +
  TOMSGQ(*) VALUES(Y N) DFT(N) MSGRPY(&REPLY)
```

This command sends an inquiry message to the display station operator (if it is used in an interactive job) or to the system operator (if it is used in a batch job). The valid replies are Y and N, and any other reply is rejected. The reply is returned in the variable &REPLY. The default translation table, QTRNTBL, is used to translate the reply to uppercase.

```
SNDUSRMSG   MSG('Enter any response to this  +
  message when ready to continue with updates.')  +
  TOMSGQ(WS01)
```

This command sends an inquiry message to a specific message queue. Any reply is valid. Because the purpose of this example is simply to wait, no CL variable is provided to receive the reply.

```
SNDUSRMSG   MSGID(USR0150)  TOUSR(FRED) MSGF(QGPL/USRMSGF)  +
  MSGDTA(&ACCTNO)  MSGTYPE(*INFO)
```

This command sends a predefined message as an information message to the message queue specified in the user profile of the specified user (FRED). The message data provided is included in the message.

STRAPF (Start Advanced Printer Function) Command

The Start Advanced Printer Function (STRAPF) command calls a menu that offers options for designing forms. From this menu you can design special symbols, the layout of a form (called a form description), print copies of the form description, or merge a spooled file with a form description and print the result.

The STRAPF command is part of the *IBM AS/400 Program Development Aid*, Program 5728-PW1. The advanced printer function uses the special printing capabilities of the IBM 5224 and 5225 work station printers. More information on the advanced printer function is in the *APF Guide*.

```
STRAPF────────────────────────────────────────────────────────

                                                 ┌──────────────┐
                                                 │Job: I  Pgm: I│
                                                 └──────────────┘
```

There are no parameters for this command.

Example

```
STRAPF
```

This command calls a menu from which you can choose options to design a symbol set and a form description.

STRAPW (Start Advanced Print Writer) Command

The Start Advanced Print Writer (STRAPW) command shows the main Advanced Print Writer (APW) menu. From this menu you can select an option that allows you to create a form description, merge the user data and the from description, display the form description list, and edit the symbols.

```
                                                           Required
STRAPW──────────────────────────────────────────────────────────────

                                                     ┌──────────────┐
                                                     │Job: I  Pgm: I│
                                                     └──────────────┘
```

There are no parameters for this command.

Example

```
STRAPW
```

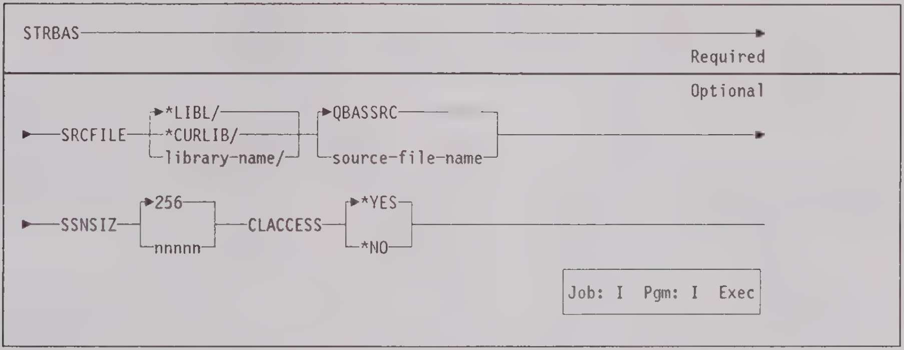
This command displays the main APW menu.

STRBAS (Start Basic) Command

The Start Basic (STRBAS) command is used to enter, change, save, or delete a BASIC program, BASIC procedures, and other data. This command permits:

- Running of source programs using the BASIC interpreter
- Compiling of BASIC programs
- Calling compiled BASIC programs
- Running of BASIC procedures.

A BASIC source program may be run without compiling it. However, compiled programs run faster than interpreted programs.



SRCFILE Parameter
Specifies the qualified name of the source file BASIC automatically uses when processing programs, commands, or procedures unless another source file is specifically requested.

The possible library values are:

<u>*LIBL:</u> The library list is used to locate the source file.

***CURLIB:** The current library for the job is used to locate the source file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the source file is located.

<u>QBASSRC:</u> Specifies the IBM-supplied BASIC source file which is the default if a source-file-name is not entered.

source-file-name: Specify the name of the source file used.

SSNSIZ Parameter
Indicates the amount of space in K (1024) bytes to reserve for the BASIC session history area.

<u>256:</u> Specifies the default value of 256 if a value for the amount of space to reserve is not entered.

nnnnn: Specify the number of K (1024) bytes of storage to reserve.

CLACCESS Parameter
Controls access to CL commands during the current BASIC session.

<u>*YES:</u> Permits access to CL commands. SYSTEM command is allowed.

***NO:** CL commands may not be used during the current session. SYSTEM command is not allowed.

Example

```
STRBAS  SRCFILE(FILELIB/BASFILE)  SSNSIZ(72)
```

This command starts a BASIC session. All programs, commands, and procedures processed during the session will use the source file BASFILE in library FILELIB. The session history will be recorded in a history file of 72 K-bytes.

STRBASPRC (Start Basic Procedure) Command

The Start Basic Procedure (STRBASPRC) command permits running a BASIC procedure that was saved as a member in a source file.

```
STRBASPRC──SRCMBR member─name──────────────────────────────────────────────►
                                                                    Required
────────────────────────────────────────────────────────────────────────────
                                                                    Optional
             ┌►*LIBL/────────┐ ┌►QBASSRC──────────┐
►──SRCFILE───┼─*CURLIB/──────┼─┤                  ├─────────────────────────►
             └─library-name/─┘ └─source-file-name─┘

           ┌►256──┐            ┌►*YES─┐
►──SSNSIZ──┤      ├──CLACCESS──┤      ├──────────────────────────────────────
           └nnnnn─┘            └─*NO──┘

                                              Job: B,I  Pgm: B,I  Exec
```

SRCMBR Parameter
Specifies the name of the member in the source file that contains the BASIC procedure run.

SRCFILE Parameter
Specifies the qualified name of the source file BASIC automatically uses when processing programs, commands, or procedures unless another source file is specifically requested.

The possible library values are:

***LIBL:** The library list is used to locate the source file.

***CURLIB:** The current library for the job is used to locate the source file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the source file is located.

QBASSRC: The IBM-supplied BASIC source file which is the default if a source-file-name is not entered.

source-file-name: Specify the name of the source file used.

SSNSIZ Parameter
Indicates the amount of space in K (1024) bytes to reserve for the BASIC session history area.

256: Specifies the default of 256 if a value for the amount of space to reserve is not entered.

nnnnn: Specify the number of K (1024) bytes of storage to reserve.

CLACCESS Parameter
Controls access to CL commands during the current BASIC session.

***YES:** Permits access to CL commands. SYSTEM command is allowed.

***NO:** CL commands may not be used during the current session. SYSTEM command is not allowed.

Example

```
STRBASPRC  SRCMBR(LEVEL1)  SRCFILE(FILELIB/BASFILE)  +
  SSNSIZ(72)
```

This command starts the BASIC procedure stored as source member LEVEL1. The source member is a member of the source file BASFILE in library FILELIB. The session history will be recorded in a history file of 72 K-bytes.

STRBGU (Start Business Graphics Utility) Command

The Start Business Graphics Utility (STRBGU) command calls the Business Graphics Utility (BGU).

```
STRBGU────────────────────────────────────────────────────────────

                                          ┌──────────────────────┐
                                          │ Job: I  Pgm: I  Exec │
                                          └──────────────────────┘
```

There are no parameters for this command.

Example

```
STRBGU
```

This command calls the Business Graphic Utility.

STRCBLDBG (Start COBOL Debug) Command

The Start COBOL Debug (STRCBLDBG) command activates the debugging code that is created when the WITH DEBUGGING MODE clause is used in a COBOL program. Ths command must be entered for each COBOL program debugged in the next COBOL run unit.

```
                                                            Optional
                   ┌►*LIBL/────────┐
STRCBLDBG───PGM────┼─*CURLIB/──────┼──program-name──────────────────
                   └─library-name/─┘

                                                   ┌──────────────────────────┐
                                                   │Job: B,I  Pgm: B,I  Exec  │
                                                   └──────────────────────────┘
```

PGM Parameter
Specifies the qualified name by which the compiled COBOL program can be known.

The possible library values are:

***LIBL:** The library list is used to locate the COBOL program.

***CURLIB:** The current library for the job is used to locate the COBOL program. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the COBOL program is located.

program-name: Specifies the name by which the COBOL program is known.

Example

```
STRCBLDBG  PGM(JAMES1/PROG7)
```

This command activates the debugging statements compiled into the program PROG7 in library JAMES1.

STRCGU (Start CGU) Command

The Start CGU (STRCGU) command calls the Character Generator Utility (CGU) main menu or allows the user to specify a specific CGU function depending on the parameter used.

```
                ┌►*SELECT─┐
                ├─1───────┤
                ├─2───────┤
STRCGU──OPTION──┼─3───────┼──────────────────────────────────────────
                ├─4───────┤
                ├─5───────┤
                └─6───────┘

                                                ┌──────────────┐
                                                │Job: I  Pgm: I│
                                                └──────────────┘
```

OPTION Parameter

This parameter calls the CGU main menu.

*SELECT: Specifies the specific CGU main menu.

1: Creates or updates user-defined double byte character set (DBCS) characters.

2: Deletes user-defined DBCS characters.

3: Prints user-defined DBCS characters.

4: Updates sort information for user-defined DBCS characters.

5: Updates the DBCS active sort table for use by the SORT function.

6: Prints a synchronization report for user-defined DBCS characters.

Example

```
STRCGU
```

This command displays the CGU main menu.

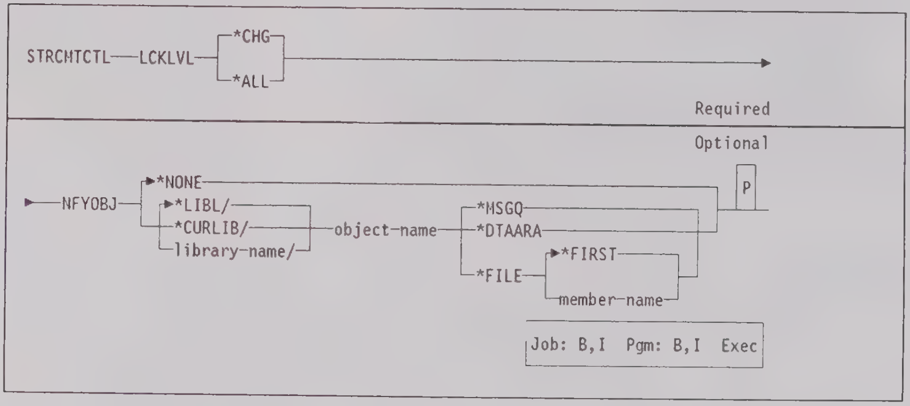
STRCMTCTL (Start Commitment Control) Command

The Start Commitment Control (STRCMTCTL) command is used to establish the commitment control environment.

You can use this command to specify the level of record locking that occurs in the commitment control environment. You can also specify a message queue, a file, or a data area to receive an entry identifying the transaction last successfully committed by the process before the system or job ends abnormally (with uncommitted changes). This entry can be specified on the Commit (COMMIT) command when the commitment boundary is established.

Before you establish a commitment control environment, you must ensure that all data base files used under commitment control are journaled to the same journal. If only after images are being journaled, the system implicitly begins journaling both before and after images for the duration of this commitment control environment.

More information on the use of journal management or the use of commitment control is in the *Backup and Recovery Guide.*



LCKLVL Parameter

Specifies the level of record locking that occurs under the commitment control environment. You can specify if all records in a file that are accessed in the commitment control environment are locked, or if only those records changed (updated, added or deleted) should be locked. Records remain locked until a commitment boundary is established or by running either the COMMIT or Rollback (ROLLBACK) command.

***CHG:** Specifies that every record read for update (in a file under commitment control) is locked. If a record is updated, added, or deleted, that record remains locked until a commitment boundary is established. Records that are accessed for update but are released without being updated are unlocked.

***ALL:** Specifies that every record accessed from files under commitment control is locked until the transaction is committed or rolled back.

NFYOBJ Parameter

Specifies the name and type of the object where notification is sent if a system or job failure occurs or a job ends with uncommitted changes. The notification text that identifies the last transaction that was successfully committed (specified in the COMMIT command) is sent only if a successful initial program load (IPL) is done after a system or job failure occurs or a job ends with uncommitted changes. The notification text is not sent if a successful COMMIT or ROLLBACK operation occurs. If at the same time there is more than one job using commitment control, each job should use a unique notify object, or the text should contain a unique identifier. If COMMIT CMTID(*NONE) is specified, this parameter is ignored.

***NONE:** Specifies that no notification is sent after an abnormal system or job end.

The possible library values are:

***LIBL:** The library list is used to locate the object.

***CURLIB:** The current library for the job is used to locate the object. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the object is located.

object-name: Specify the qualified name of the object to receive notification of the last transaction that is successfully committed (library-name object-name). You must have the required authority for the object specified.

***MSGQ:** Specifies that the text identifying the last commitment boundary is placed on the specified message queue.

***DTAARA:** Specifies that the text identifying the last commitment boundary is placed in the specified data area. The data area specified must be of type character, and unique to this job. The text is padded or truncated to fit the data area.

***FILE:** Specifies that the text identifying the last commitment boundary is added to the specified physical file.

<u>*FIRST:</u> Specifies that the first member of the physical file receives the notification.

member-name: Specify the name of the member of the physical file that receives the notification.

Examples

```
STRCMTCTL   LCKLVL(*CHG)
```

This command specifies that only records that are updated, inserted, or deleted are locked until the next commitment boundary is established. No identification for the commitment boundary is sent after the initial program load (IPL) following an abnormal system end or abnormal job end.

```
STRCMTCTL   LCKLVL(*ALL)  NFYOBJ(RCVLIB/MYFILE *FILE IDSAVE)
```

This command specifies that all records accessed in files under commitment control used by the program are locked until the next commitment boundary is established. If an abnormal system end or abnormal job end occurs, the text or characters used to identify the commitment boundary (as specified in the COMMIT command) are placed in the member named IDSAVE of file MYFILE in the RCVLIB library.

STRCPYSCN (Start Copy Screen) Command

The Start Copy Screen (STRCPYSCN) command allows you to copy the screens of another display station on your display station to observe what is happening and diagnose problems.

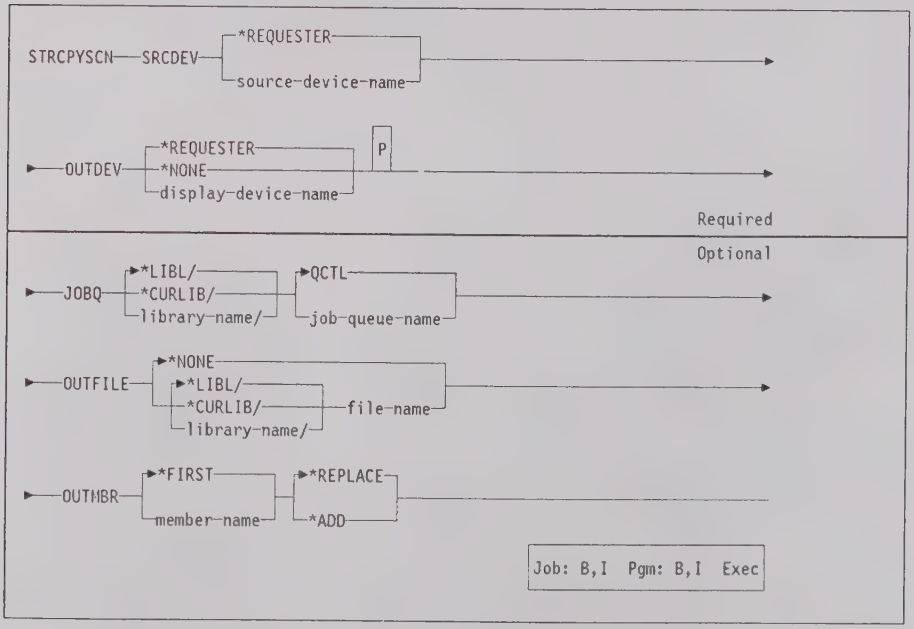
If STRCPYSCN is started to copy screens from a source device that has the wide-screen feature to an out device with a regular-width screen, the command is accepted, but wide-screen images are not shown and an informational message is sent to the target work station saying that the screen image was not displayed.

If STRCPYSCN is started to copy screens from a source device that has the graphics feature, the command is accepted and character information is shown, but graphics information is not shown. No message is sent.

If the out device is not the requesting device, then the out device cannot be signed on. If the out device is signed on, a message is sent to the requester saying that the device is not available for copying. If the source device is signed off after screen copy has begun, the function automatically ends.

Note: The copy screen function can be ended by either the source or target device.

Restrictions: (1) Permission must be given from the user of the source work station. (2) When a request is made to begin screen image copying, a break message is sent to the user of the source work station to inform him that the screens are going to be copied. The user must reply to this message before any screens are copied. (3) *REQUESTER is not valid for the SRCDEV or OUTDEV parameters when the command is submitted to batch.



SRCDEV Parameter

Specifies the display station that is used as the source for the screen images being copied.

***REQUESTER:** Specifies that the screens are copied from the display station that issued this command.

source-device-name: Specify the display station name (other than the *REQUESTER), that is having its screens copied.

OUTDEV Parameter

Specifies the output device for the copying process.

***REQUESTER:** Specifies that screens are copied to the work station that this command is issued from. If SRCDEV is *REQUESTER, then OUTDEV(*REQUESTER) cannot be specified.

***NONE:** Specifies that the copied screens do not go to a display station. If OUTDEV is *NONE, then OUTFILE must be specified.

display-device-name: Specify the name of the display station (other than the *REQUESTER) that displays the copied screens.

JOBQ Parameter

Specifies the job queue used to submit the job which shows the screens from the source device on the target device when the requesting device is not the target device. When OUTDEV(*REQUESTER) is specified, this parameter is ignored, since it defaults to the values for the target display station and then a submit job is not necessary.

The possible library values are:

***LIBL:** The library list is used to locate the job queue.

***CURLIB:** The current library for the job is used to locate the job queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the job queue is located.

QCTL: Specifies that the default job queue, QCTL is used. The library list is used to locate the job queue.

job-queue-name: Specify the qualified name of the job queue where the job of processing the copied screens is submitted.

OUTFILE Parameter

Specifies the qualified name of the data base file where the copied screens are directed. If the OUTFILE does not exist, this command creates a data base file in the specified library. Screens can be copied to a data base file instead of or in addition to being shown on another work station, thereby keeping a record of the screens that were copied and shown.

***NONE:** Specifies that the output is not directed to a data base file.

The possible library values are:

***LIBL:** The library list is used to locate the file.

***CURLIB:** The current library for the job is used to locate the file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the file is located.

file-name: Specifies the qualified name of the data base file that is to receive the output of the display.

If you qualify the data base file name with *LIBL, but the system cannot find the file, one is created in the QGPL library. Or if you specify a library name, it is created in the specified library.

OUTMBR Parameter

Specifies the name of the data base file member where the output of the display is directed.

***FIRST:** Specifies that the first member in the file is to receive the output. If OUTMBR(*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified in the OUTFILE parameter.

member-name: Specify the member to receive the output. If the specified member name does not exist, the system creates it.

***REPLACE:** The new data is added to the file member in place of the existing data.

***ADD:** The new data is added to the end of the data already in the file member.

Example

```
STRCPYSCN  SRCDEV(WS2)  OUTDEV(*REQUESTER)
```

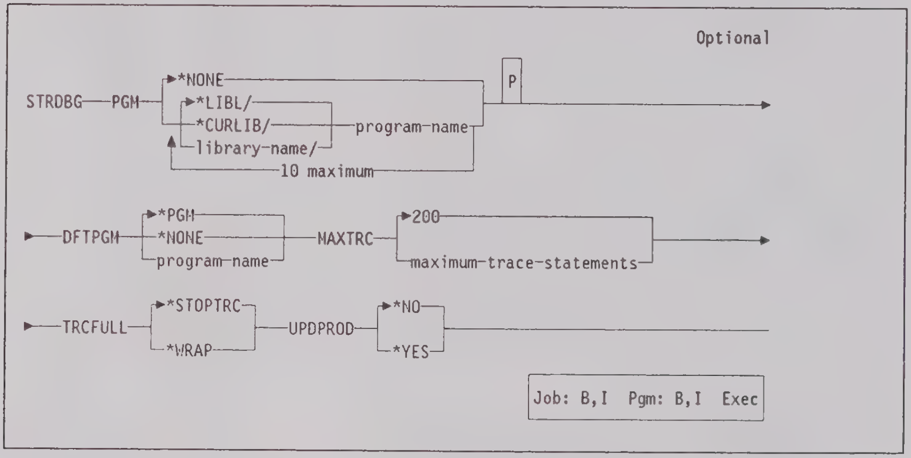
This command sends an inquiry message to the user of work station, WS2. The message indicates that the display station screens are about to be copied to another display station. If the user of that display does not wish this to happen, then a cancel (N) reply stops the operation from beginning. To allow the operation to begin, the user responds Y to the message.

STRDBG (Start Debug) Command

The Start Debug (STRDBG) command puts a job into debug mode and, optionally, adds as many as ten programs to debug mode. It also specifies certain attributes of the debugging session. For example, it can specify whether data base files in production libraries can be updated while in debug mode.

The Change Debug (CHGDBG) command can be used later in the job to change the attributes of the debug mode. Also, programs can be added to or removed from the debugging session if they are specified in the Add Program (ADDPGM) or Remove Program (RMVPGM) commands.

Restriction: This command is not valid in debug mode. To end debug mode, refer to the ENDDBG (End Debug) command.



PGM Parameter
Specifies the qualified names of up to ten programs to debug in the job. Before a program can be debugged, its name must be specified here or in the ADDPGM command.

***NONE:** No program names are specified at the start of the debugging session. The ADDPGM command can be used to add programs later.

The possible library values are:

***LIBL:** The library list is used to locate the program.

***CURLIB:** The current library for the job is used to locate the program. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the program is located.

program-name: Specify the qualified names of one to ten programs to debug. You cannot debug two programs that have the same name at the same time.

DFTPGM Parameter
Specifies the name of the program to use as the default program during debug mode. The program specified here is used as the default program for any of the other debug commands that specify *DFTPGM on their PGM parameter. (That is, if a default program was previously specified, this parameter can change it.)

***PGM:** The program named in the PGM parameter of *this* command is the default program for the job's debugging environment. If PGM has more than one program name specified, the first program named in the list is the default program. If PGM(*NONE) is specified or is the default, DFTPGM(*NONE) is also assumed when *PGM is specified here.

***NONE:** No program is specified as the default program; if a program was specified as a default program, it is no longer the default program. If the job has no default program, *DFTPGM cannot be specified on the PGM parameter of any other debug commands.

program-name: Specify the name of the program to use as the default program during debug mode. The same name (in qualified form) must also be specified in the PGM parameter of this command.

MAXTRC Parameter
Specifies the maximum number of trace statements that the system puts into the job's trace file before either stopping, tracing, or wrapping around (over-

laying) on the trace file. When the trace file contains the maximum specified, the system performs the actions specified in the TRCFULL parameter.

Note: Instruction stepping can be performed on a program being debugged in an interactive environment by setting the maximum number of trace statements to 1 and the TRCFULL parameter to *STOPTRC.

200: Specifies a maximum of 200 trace statements can be put into the trace file before tracing is ended or wrapping on the file occurs.

maximum-trace-statements: Specify the maximum number of trace statements that can be in the trace file.

TRCFULL Parameter

Specifies what happens when the job's trace file is full (that is, it contains the maximum number of trace statements specified by the MAXTRC parameter).

***STOPTRC:** In a batch environment, tracing stops but the program continues processing. In an interactive environment, control is given to the user when a breakpoint occurs. If the user continues processing, a breakpoint occurs before processing each subsequent statement within the range of statements being traced, and the trace file is extended to contain the new entry.

***WRAP:** The trace file is overlaid with new trace statements as they occur, wrapping from the beginning of the file. The program will complete processing with no message to indicate that wrapping has occurred. The trace file never has more than the maximum specified statements, and they are the most recently recorded statements.

UPDPROD Parameter

Specifies whether or not data base files in a production library can be opened for updating records, or for adding new records, while the job is in debug mode. If not, the files must be copied into a test library before trying to run a program that uses the files.

***NO:** Data base files in production libraries cannot be updated while the job is in debug mode. Data base files can be opened for reading only. This protects data base files from unwanted updates while a program is being debugged. The exception to this is starting debug mode after a production library is already opened. (This is possible if the file is opened with SHARE(*YES) in a different program first.) To prevent this, start debug before any files are opened, or use SHARE(*NO) on the Override Data Base File (OVRDBF) command for all files used in the programs being debugged.

***YES:** Data base files in production libraries can be updated while the job is in debug mode.

Example

```
STRDBG  PGM(TESTLIB/PAYROLL)
```

This command starts debug mode to debug the program PAYROLL, which is in the test library TESTLIB. If tracing is used, a maximum of 200 trace statements can be stored in the trace before tracing stops. Any data base files updated by the PAYROLL program must be in a test library.

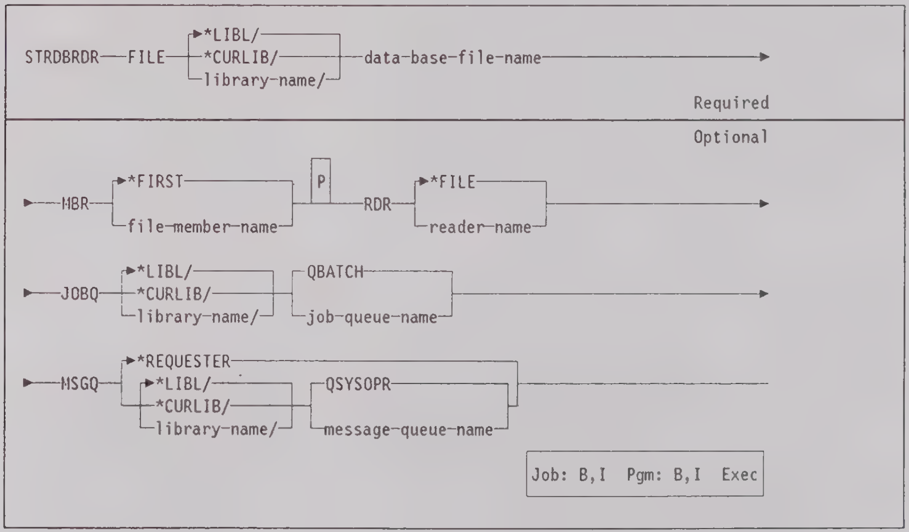
STRDBRDR (Start Data Base Reader) Command

The Start Data Base Reader (STRDBRDR) command starts a spooled reader to a data base file; the reader reads a batch input stream from the data base and places the jobs onto one or more job queues. This command specifies the name of the data base file and member from which the input stream is read, the name of the reader, and the names of the job queue and message queue that are used.

More than one reader can be active at the same time (as determined by the spooled subsystem description). Each data base reader must have a unique reader name, and the specified file or member must be available. The reader can also be held or canceled if the Hold Reader (HLDRDR) or End Reader (ENDRDR) command is used.

Because each reader runs independently of the job that started it, the user can continue doing other work on the system after he has started a reader. The reader is owned by the user who issues the STRDBRDR command.

Restriction: The specified data base file must consist of single field records and must have an arrival sequence access path, or it must be a standard data base source file.



FILE Parameter

Specifies the qualified name of the data base file that the input stream is to be read from. The file must be available for allocation to the spooled reader before the reader can be started.

The possible library values are:

***LIBL:** The library list is used to locate the file.

***CURLIB:** The current library for the job is used to locate the file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the file is located.

MBR Parameter

Specifies the name of the member in the specified file that contains the input stream that is read.

***FIRST:** No member name is specified; the first member in the file is used.

file-member-name: Specify the name of the member that contains the input stream that is read by the reader.

RDR Parameter

Specifies the name of the spooled reader being started. Each reader name must be unique.

***FILE:** The name of the reader is the same as the name of the data base file specified by the FILE parameter.

reader-name: Specify the name that the reader being started is identified by.

JOBQ Parameter

Specifies the qualified name of the job queue where the spooled reader places jobs. An entry is placed on this job queue for each job in the input stream that specifies JOBQ(*RDR) on its Batch Job

(BCHJOB) command. If *RDR is not specified on the BCHJOB command, the job queue specified in the BCHJOB command or in the job description is used. (Note that the job queue for each job in the input stream can be different.)

Restriction: If the user starting the reader is not authorized for the job queue used by the reader for a job in the input stream, an error message is sent to the message queue specified by MSGQ and the job is not run.

The possible library values are:

***LIBL:** The library list is used to locate the job queue.

***CURLIB:** The current library for the job is used to locate the job queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the job queue is located.

QBATCH: The job entry is to be placed on the QBATCH job queue. QBATCH is the default if JOBQ(*RDR) is specified on the BCHJOB command.

job-queue-name: Specify the qualified name of the job queue that each job read by this reader is sent to if JOBQ(*RDR) is specified on its BCHJOB command.

MSGQ Parameter

Specifies the qualified name of the message queue that the messages created by the reader are sent to.

QSYSOPR: The messages are sent to the system operator's message queue, QSYSOPR.

The possible library values are:

***LIBL:** The library list is used to locate the message queue.

***CURLIB:** The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message queue is located.

***REQUESTER:** The messages are to be sent to the message queue of the user who started the reader. This value is not valid for batch jobs.

message-queue-name: Specify the qualified name of the message queue that any messages created by the reader are sent to.

Example

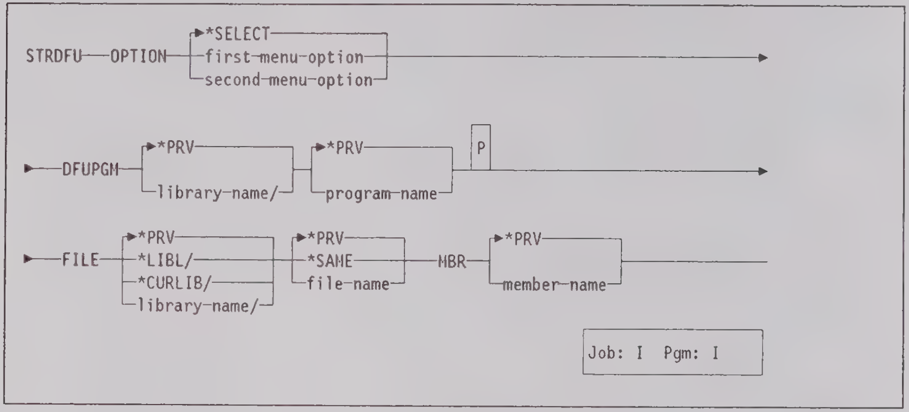
```
STRDBRDR  FILE(QGPL/BILLING)
```

This command starts a spooled reader that reads its input from the data base file named BILLING, which is in the QGPL library. The reader name is also BILLING because the RDR parameter was not specified. The first member in the BILLING file contains the input stream to be processed. The default job queue QBATCH and the message queue QSYSOPR are used by the data base reader.

STRDFU (Start DFU) Command

The Start Data File Utility (STRDFU) command starts the Data File Utility (DFU).

Restriction: Any overrides on the specified file are ignored.



OPTION Parameter
Specifies the option to use as a value for the DFU main menu.

***SELECT:** The DFU main menu appears. You can select an option from the menu.

first-menu-option: Specify a number between 1 and 5. Specify 1 to run, 2 to create, 3 to change, or 4 to delete a DFU program. Specify 5 to run a temporary DFU program.

second-menu-option: If you specify 1 (to run a DFU program), you can also specify a second option. The second option must be 1 or 2. Specify 1 to change data, or 2 to display data without changing it.

DFUPGM Parameter
Specifies the name of the DFU program to run, create, change, or delete. You do not use this parameter if you select option 5 (to run a temporary DFU program).

***PRV:** DFU uses the program that was used in your last DFU session.

library-name: Specify the library where the DFU program is located.

***PRV:** DFU uses the program that was used in your last DFU session.

program-name: Specify the name of the DFU program used.

FILE Parameter
Specifies the qualified name of the data base file you want to change or display.

***PRV:** DFU uses the file that was used in your last DFU session.

The possible library values are:

***LIBL:** The library list is used to locate the data file.

***CURLIB:** The current library for the job is used to locate the data file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the data file is located.

***PRV:** DFU uses the file that was used in your last DFU session.

***SAME:** DFU uses the file that was used to define the program. You can specify *SAME only if you select option 1 (to run a DFU program) or option 3 (to change a DFU program).

file-name: Specify the qualified name of the data file you want DFU to process.

MBR Parameter
Specifies the member in the file you want to change or display.

***PRV:** DFU uses the member that was used in your last DFU session.

member-name: Specify the name of the member you want DFU to process.

Example

```
STRDFU
```

This command starts the Data File Utility (DFU).

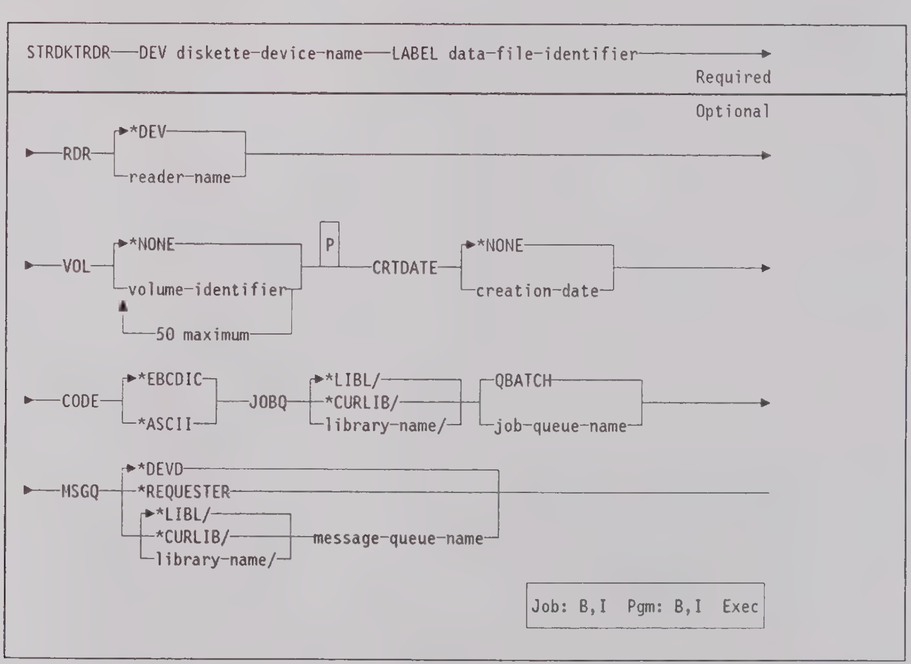
STRDKTRDR (Start Diskette Reader) Command

The Start Diskette Reader (STRDKTRDR) command starts a spooled reader to the specified diskette device to read a batch input stream into the system. This command specifies the name of the diskette device that the input stream is read from; the volume and data file that the input stream is on; and the names of the job queue and message queue to be used.

More than one reader, which is a system job, can be active at the same time (as determined by the spooled subsystem description). Each reader must have a unique reader name and have its own input device assigned to it. A reader that has been started can be actively reading input or waiting for device input. The reader can also be held or ended if the Hold Reader (HLDRDR) or End Reader (ENDRDR) command is used. The reader is ended at end-of-file.

Because each reader runs independently of the job that started it, the user can continue doing other work on the system after he has started a reader. The reader is owned by the user who issues the STRDKTRDR command.

Restriction: This command cannot be used to read data files of diskettes that are in the save/restore format.



DEV Parameter

Specifies the name of the diskette device that is used to read in an input stream.

LABEL Parameter

Specifies the data file identifier of the file that is processed as an input stream. (For an expanded description of the LABEL parameter, see Appendix A, "Expanded Parameter Descriptions" in Volume 1.)

RDR Parameter

Specifies the name of the spooled reader that is started. Each reader name must be unique.

***DEV:** The name of the reader is the same as the name of the diskette device specified in the DEV parameter.

reader-name: Specify the name that the reader being started is identified by.

VOL Parameter

Specifies the volume identifiers of the diskette volumes that contain the input stream that is processed. The volumes must be put on the device in the same order that the volume identifiers are specified in this parameter. (For an expanded description of the VOL parameter, see Appendix A, "Expanded Parameter Descriptions" in Volume 1.)

***NONE:** No volume identifier is specified. Use the current volume.

volume-identifier: Specify the identifiers of one or more volumes in the order that they are put on the device and read. Each identifier can have 6 alphameric characters or less. A blank is used as a separator character when listing multiple volumes.

CRTDATE Parameter

Specifies when the diskette data file was created on diskette; the creation date should not be specified if the date is not checked. If the date written on the diskette containing the data file does not match the date specified here, an error message is sent to the message queue named in the MSGQ parameter.

***NONE:** The creation date is not specified; no check is made.

creation-date: Specify the creation date of the data file that is read. The date should be in the format specified by the QDATFMT system value.

CODE Parameter

Specifies the type of character code used to read the diskette data into the job queue.

***EBCDIC:** The diskette data file is written in the EBCDIC character code.

***ASCII:** The diskette data file is written in the ASCII character code.

JOBQ Parameter

Specifies the qualified name of the job queue that the spooled reader places job entries on. An entry is placed on the queue specified here only for each job in the input stream that specifies JOBQ(*RDR) on its Batch Job (BCHJOB) command. If *RDR is not specified on the BCHJOB command, the job queue specified here is ignored and the job queue named in the BCHJOB command or in the job description is used. (Note that, if *RDR is not specified, the job queue for each job in the input stream can be different.)

Restriction: If the user starting the diskette reader is not authorized for the job queue used by the reader for a job in the input stream, an error message is sent to the message queue specified by MSGQ and the job is not run.

The possible library values are:

***LIBL:** The library list is used to locate the job queue.

***CURLIB:** The current library for the job is used to locate the job queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the job queue is located.

QBATCH: The job entry is placed on the QBATCH job queue.

job-queue-name: Specify the qualified name of the job queue to which each job read by this reader is sent if JOBQ(*RDR) is specified on its JOB command.

MSGQ Parameter

Specifies the qualified name of the message queue that messages created by the diskette reader are to be sent to.

***DEVD:** The messages are sent to the message queue specified in the device description of the device named in the DEV parameter.

***REQUESTER:** The messages are sent to the message queue of the user who started the reader. This value is not valid for batch jobs.

The possible library values are:

***LIBL:** The library list is used to locate the message queue.

***CURLIB:** The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message queue is located.

message-queue-name: Specify the qualified name of the message queue that the diskette reader messages created by the reader are sent to.

Example

```
STRDKTRDR  DEV(QDKT)  LABEL(OCT24)  VOL(SALES)
```

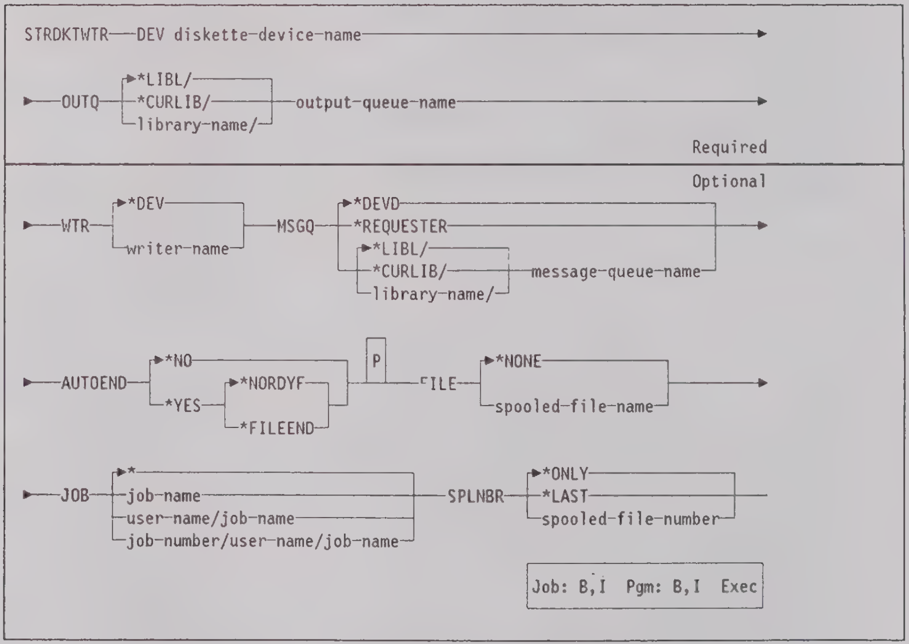
This command starts the spooled reader named QDKT, which reads diskette input from the device QDKT. (Because *DEV was the default on the unspecified RDR parameter, the device name QDKT is also used as the reader name.) The reader reads its input from the data file named OCT24 whose volume identifiers must be SALES. The default job queue QBATCH and the message queue QSYSOPR are used by the diskette reader.

STRDKTWTR (Start Diskette Writer) Command

The Start Diskette Writer (STRDKTWTR) command starts a spooling writer to the specified diskette device. The writer, which is a system job, takes spooled output files from an output queue and produces (writes) the output on the diskette device. This command specifies the names of the diskette device and the writer and the names of the output and message queues to be used.

More than one writer can be active at the same time (as determined by the spooling subsystem description). Each writer must have a unique writer name, its own output queue, and its own device. A writer that has been started can be actively writing output or waiting for a file to be put on the output queue. Optionally, the writer can be automatically stopped when it has processed all the files on the output queue. The writer can also be held or stopped if the Hold Writer (HLDWTR) or End Writer (ENDWTR) command is used.

Because each writer runs independently of the job that started it, the user can continue doing other work on the system after he has started a writer.



DEV Parameter
Specifies the name of the diskette device that the spooled output is written to.

OUTQ Parameter
Specifies the qualified name of the output queue from which the writer processes spooled output files. The output queue must be available before the writer can be started.

The possible library values are:

*LIBL: The library list is used to locate the output queue.

*CURLIB: The current library for the job is used to locate the output queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the output queue is located.

WTR Parameter
Specifies the name of the writer being started. Each writer name must be unique.

*DEV: The name of the writer is to be the same as the name of the diskette device specified in the DEV parameter.

writer-name: Specify the name that the writer being started is identified by.

MSGQ Parameter
Specifies the qualified name of the message queue that messages created by the diskette writer are sent to.

***DEVD:** The messages are sent to the message queue specified in the device description of the device named in the DEV parameter.

***REQUESTER:** The messages are sent to the message queue of the user who started the writer. This value is not valid for batch jobs.

The possible library values are:

***LIBL:** The library list is used to locate the message queue.

***CURLIB:** The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message queue is located.

message-queue-name: Specify the qualified name of the message queue that the diskette writer messages are sent to.

AUTOEND Parameter

Specifies whether the writer stops automatically when the output queue has no more available entries for spooled output files to be written to the diskette device.

***NO:** The writer does not end when the last available entry has been removed from the output queue; it waits for another spooled file entry to be put on the queue.

***YES:** The writer automatically ends after it has reached the state specified by the second part of the parameter (*NORDYF or *FILEEND) files have been removed from the output queue.

***NORDYF:** The writer automatically ends when there are no ready files (all the available entries have been removed from the output file).

***FILEEND:** The writer stops after it has finished processing one spooled output file.

FILE Parameter

Specifies the name of the first (or only) spooled output file processed by the spooling writer and written to diskette. If several files are available on the output queue, the next file produced is the first one available with the highest priority.

***NONE:** No spooled file name is specified; the first spooled file that becomes available on the output queue is processed first.

spooled-file-name: Specify the name of the spooled output file that is the first (or only) output file to be written to diskette.

JOB Parameter

Specifies the name of the job that created the spooled file being written to diskette. This parameter is valid only if a spooled file name is specified in the FILE parameter.

***:** The job that issued this STRDKTWTR command is the job that created the spooled file.

job-name: Specify the name of the job that created the spooled file. If no job qualifier is given, all of the jobs currently in the system are searched for the simple job name. (For an expanded description of the JOB parameter and duplicate job names, see Appendix A, "Expanded Parameter Descriptions" in Volume 1.)

SPLNBR Parameter

Specifies the number of the spooled output file that is processed first. This parameter is valid only if a spooled file name is specified in the FILE parameter. (For an expanded description of the SPLNBR parameter, see Appendix A, "Expanded Parameter Descriptions" in Volume 1.)

***ONLY:** Only one spooled output file from the job on the specified output queue has the specified name; no spooled file number is needed. If *ONLY is specified and more than one spooled file on the output queue has the specified name, an error message is displayed to the user.

***LAST:** The highest-numbered spooled output file with the specified file name from the job on the specified output queue is the file processed first.

spooled-file-number: Specify the number of the specified file from the job on the specified output queue that is processed first.

Example

```
STRDKTWTR  DEV(QDKT)  OUTQ(QDKT)  AUTOEND(*YES)
```

This command starts a spooling writer to the diskette drive. The files written on the diskettes are on the IBM-supplied output queue QDKT. When all the files have been written (no more entries are on the QDKT output queue), the writer is automatically ended and the diskette drive is available for other uses.

STRDSKCOL (Start Disk Data Collection) Command

The Start Disk Data Collection (STRDSKCOL) command initiates trace data collection for storage management for disk activity logging. Once tracing is started, a trace record is created for every disk I/O request.

After the data has been collected, the End Disk Data Collection (ENDDSKCOL) command is used to turn tracing off and optionally print a report from the accumulated data. The Print Disk Activity Report (PRTDSKRPT) command may subsequently be used to generate reports from the same data.

The ENDDSKCOL command is part of the IBM AS/400 Performance Tools Product, Program 5728-PT1. More information on the performance tools is in the *Performance Tools Guide*.

```
STRDSKCOL──────────────────────────────────────────────────────►
                                                        Required
─────────────────────────────────────────────────────────────────
                                                        Optional
            ┌►*WRAP────┐
►──TRCFULL──┤          ├─────────────────────────────────────────
            └─*STOPTRC─┘

                                               Job: B,I  Pgm: B,I
```

TRCFULL Parameter
Specifies the action to take when the internal trace buffer fills up.

***WRAP:** Continue the trace, overwriting the oldest data in the buffer with the new data.

***STOPTRC:** Stop the trace.

Examples

```
STRDSKCOL
```

This command starts disk activity data collection. If the trace buffer fills up, the new data overwrites the oldest data, so the most recent data is always available.

```
STRDSKCOL  TRCFULL(*STOPTRC)
```

This command starts disk activity data collection. If the trace buffer fills up, the trace is stopped.

STREDU (Start Education) Command

The Start Education (STREDU) command starts the education session.

User classes *USER, *SYSOPR, *PGMR, *SECADM and *SECOFR may use this command. *SECADM and *SECOFR user classes must have administrator authority.

```
STREDU──────────────────────────────────────────────────────

                                              Job: I  Pgm: I
```

There are no parameters for this command.

Example

```
STREDU
```

This command shows the following menus:

The `Start Education Administration` menu is shown for the Administrator.

The `Introduction to Online Education` menu is shown for the new student that was enrolled by the Administrator.

The `Specify your Name` menu is shown for the new student that was not enrolled.

The `Select Course Option` menu is shown for the enrolled student.

STREML3270 (Start 3270 Display Emulation) Command

The Start 3270 Display Emulation (STREML3270) command starts a 3270 display emulation session. You can enter this command from your display device on the command line or from any 1920-character display station that allows you to enter CL commands. This command can also be entered from a batch job using the DSPDEV parameter on this command.

The STREML3270 command can be in a CL program, or it can be in the first program in a user profile and can be run when you sign on your display device.

More information on device emulation is in the *3270 Device Emulation User's Guide.*

```
          (1) ┌─EMLCTL─emulation─controller─description─name─┐ ┌─┐
STREML3270────┼─EMLDEV─emulation─device─description─name─────┤ │P│──────────────────►
              └─EMLLOC─emulation─location─name───────────────┘ └─┘
                                                                          Required
──────────────────────────────────────────────────────────────────────────────────
                                                                          Optional
          (2) ┌►*CURRENT──────────────┐                ┌►120──────────────┐
►──DSPDEV─────┤                       ├──INZWAIT───────┼─*NOMAX───────────┼──────►
              └─display─device─name───┘                └─number─of─seconds┘

              ┌►*EMLDEV─┐           (3) ┌►*LCL──────────────────────────┐
►──NUMLCK─────┼─*NO─────┼──KBDTYPE──────┼─*TRNTBL───────────────────────┼────────►
              └─*YES────┘               └─display─keyboard─identifier───┘

              ┌►*PA2──────┐                  ┌►*PA1──────┐
              ├─*NONE─────┤                  ├─*NONE─────┤
              ├─*F─key────┤                  ├─*F─key────┤
►──PAGEUP─────┼─*PA1──────┼──PAGEDOWN────────┼─*PA2──────┼────────────────────────►
              ├─*PA3──────┤                  ├─*PA3──────┤
              ├─*CLEAR────┤                  ├─*CLEAR────┤
              └─*ERASEINP─┘                  └─*ERASEINP─┘

                ┌►*DFT──────┐
►──TESTREQ──────┼─*CLEAR────┼─────────────────────────────────────────────────────►
                └─*ERASEINP─┘

      (4)       ┌►*NO──┐         (4)          ┌►*NONE──────────────────────────────────┐
►──IGCEMLPC─────┤      ├───EMLPRTDEV──────────┼─emulation─device─description─name──────┼──►
                └─*YES─┘                      ├─*EMLCTL────────────────────────────────┤
                                              └─*EMLLOC────────────────────────────────┘

                 ┌►*KBDTYPE───────────────────────────────────────────┐
►──TRNTBLOUT─────┤  ┌─*LIBL/─────────┐                                 ├──────────►
                 └──┼─*CURLIB/───────┼──user─defined─outgoing─table────┘
                    └─library─name/──┘
(continued on next page)
```

(continued from previous page)

```
              ┌►*KBDTYPE──────────────────────────────────────────┐
►──TRNTBLIN───┤  ┌─*LIBL/────────┐                                ├──────────────
              └──┼─*CURLIB/──────┼──user-defined-incoming-table───┘
                 └─library-name/─┘

(1) Either EMLCTL, EMLDEV, or EMLLOC is required.
(2) DSPDEV is required when the command is a batch job.  DSPDEV is ignored
    when the command is an interactive job.
(3) This is required for remote display devices and is ignored for local
    display devices, except if *LCL or *TRNTBL is specified.
(4) DBCS systems only.

                                                  Job: B,I  Pgm: B,I  Exec
```

EMLCTL Parameter

Specifies the name of a BSC controller description (APPTYPE(*EML)) or SNA controller description type of HOST that has attached 3270 emulation display devices. When this parameter is specified, the requesting display device is linked to an available 3270 emulation display device on the emulation controller. The controller of the device must be available, and the requester of the command must be authorized to use the controller.

EMLDEV Parameter

Specifies the name of a BSC or an SNA emulation display device that is linked to the requesting display device to emulate a 3270 display device. The requester must be authorized to this device and the device must be available.

EMLLOC Parameter

Specifies the emulation remote location name that describes the location of the 3270 emulation display devices. When this parameter is specified, the requesting display device is linked to an available 3270 display device referred to by the location. At least one emulation device referenced by the location must be available, and the requester of the command must be authorized to use the device.

DSPDEV Parameter

Specifies the name of the display device used for display emulation when the command is in a batch job. The 3270 support tries to acquire the display device by this name; if the display device is acquired, the 3270 display emulation is active on that display device.

***CURRENT:** Specifies that the current display device name is used for display emulation.

display-device-name: Specify the display device name used for display emulation.

INZWAIT Parameter

Specifies the first amount of time that the utility waits for the first display from the host system. This parameter only applies to the first read to the host system. The value is the number of seconds. If the host system does not send the first display in this time, the emulation session is ended and message CPF8508 is returned to the requester.

120: The utility waits 120 seconds for the first display from the host system.

***NOMAX:** There is no limit on the amount of time the utility waits for the first display from the host system. This value can be used when you are not sure when the host system is active to this session. You can end the request by using the system request and ending request functions.

number-of-seconds: Specify the amount of time the utility waits for the first display from the host system. The range of allowed values is from 1 through 32767.

NUMLCK Parameter

Specifies if the numeric shift lock is set automatically on the 5250 data entry display device for numeric input fields received from the host system.

***EMLDEV:** Numeric shift lock is specified in the emulation device description in the EMLNUMLCK keyword.

***NO:** Numeric input fields are displayed on the 5250 with the keyboard shift at the lower setting.

***YES:** Numeric input fields are displayed on the 5250 with the data entry keyboard numeric shift locked to the upper setting. The operator can press the ALPHA key on the data entry keyboard to reset the numeric shift to the lower position. NUMLCK(*YES) is ignored for display devices with typewriter keyboards configured.

KBDTYPE Parameter

Specifies, for remote display devices only, the type of keyboard used on the display device.

This translation table does not apply to 3270 display stations running SNA display emulation, thus is ignored for 3270 data stream pass-through.

***LCL:** The display device that requested 3270 display emulation is a local display device. The type of keyboard used is determined from the display device description.

***TRNTBL:** Allows user-defined translate tables to be used. The character translation is defined in the translation tables specified by the TRNTBLOUT and TRNTBLIN parameters.

display-keyboard-identifier: Specify the 3-character identifier that specifies the language group that displays emulation. The identifier specified consists of 3 characters (zzz), as follows:

Character Translation Table

Language/Country	Identifier
Arabic X/Basic	CLB
Austria/Germany	AGB
Austria/Germany Multinational	AGI
Belgium Multinational	BLI
Canadian French	CAB
Canadian French Multinational	CAI
Cyrillic	CYB
Denmark	DMB
Denmark Multinational	DMI
Finland/Sweden	FNB
Finland/Sweden Multinational	FNI
France (Azerty)	FAB
France (Azerty) Multinational	FAI
France (Qwerty)	FQB
France (Qwerty) Multinational	FQI
Greece	GKB
Hebrew	NCB
Iceland	ICB
Iceland Multinational	ICI
International	INB
International Multinational	INI
Italy	ITB
Italy Multinational	ITI
Japan English	JEB
Japan English Multinational	JEI
Japan Kanji	JKB (For PS/55 and 5295 display stations)
Japan Katakana	KAB (For 5251, 5291, 5292, and 3180 Katakana display stations)
Korea	KOB
Netherlands	NEB
Netherlands Multinational	NEI
Norway	NWB
Norway Multinational	NWI
Simplified Chinese	RCB
Portugal	PRB
Portugal Multinational	PRI
Latin 2	ROB
Spain	SPB
Spain Multinational	SPI
Spanish Speaking	SSB
Spanish Speaking Multinational	SSI
Sweden	SWB
Sweden Multinational	SWI
Switzerland/French Multinational	SFI
Switzerland/German Multinational	SGI
Traditional Chinese	TAB
Thailand	THB
Turkey	TKB
United Kingdom	UKB
United Kingdom Multinational	UKI
United States/Canada	USB
United States/Canada Multinational	USI
Yugoslavia	YGI

Note: For example, KBDTYPE(USB) indicates a keyboard using the basic United States/Canada character set.

PAGEUP Parameter

Assigns another function to the Page Up (Roll Down) key on the 5250 keyboard when 3270 display emulation is active. This assignment is in effect when the number is not larger than the maximum number of input fields.

***PA2:** Specifies the 3270 PA2 function assigned to the Page Up key.

***NONE:** Specifies no 3270 function is assigned to the Page Up key. When there are fewer input fields on the display than allowed by the 5250, the Page Up key has no function.

***F-key:** Specifies the 3270 F function assigned to the Page Up key.

***PA-key:** Specifies the 3270 PA function assigned to the Page Up key.

***CLEAR:** Specifies the 3270 Clear function assigned to the Page Up key.

***ERASEINP:** Specifies the 3270 Erase Input function assigned to the Page Up key.

PAGEDOWN Parameter

Assigns another function to the Page Down (Roll Up) key on the 5250 keyboard when 3270 display emulation is active. This assignment is in effect when the number is not larger than the maximum number of input fields.

***PA1:** Specifies the 3270 PA1 function assigned to the Page Down key.

***NONE:** No 3270 function is assigned to the Page Down key. When there are fewer input fields on the display than allowed by the 5250, the Page Down key has no function.

***F-key:** Specifies the 3270 F function assigned to the Page Down key.

***PA-key:** Specifies the 3270 PA function assigned to the Page Down key.

***CLEAR:** Specifies the 3270 Clear function assigned to the Page Down key.

***ERASEINP:** Specifies the 3270 Erase Input function assigned to the Page Down key.

TESTREQ Parameter

Assigns another function to the Test Request key on the 5250 keyboard when 3270 display emulation is active.

***DFT:** Normal function is assigned to the Test Request key. This is the system default. The normal function is defined in the emulation Help text, and depends on if the 3270 emulation display device is *BSCT or *PLU1. BSC defaults to a 3270 test request function, while SNA defaults to a 3270 system request function.

***CLEAR:** The Clear key is assigned to the Test Request key.

***ERASEINP:** The Erase Input key is assigned to the Test Request key.

IGCEMLPC Parameter

SNA Japanese 3270 PC emulation or 3270 device emulation is used.

***NO:** SNA Japanese 3270 PC emulation is not used.

***YES:** SNA Japanese 3270 PC emulation is used.

EMLPRTDEV Parameter

Specifies the printer device used for SNA Japanese 3270 PC emulation. The printer device is selected after the display device is selected.

***NONE:** SNA Japanese 3270 PC emulation with display emulation is not being used. No emulation printer device is selected.

emulation-device-description-name: Specify the emulation printer device with the selected emulation display device for SNA Japanese 3270 PC emulation.

***EMLCTL:** The first available emulation printer device from the specified emulation controller in the EMLCTL parameter is used.

***EMLLOC:** The first available emulation printer device from the specified emulation location in the EMLLOC parameter is used.

TRNTBLOUT Parameter

Specifies the outgoing translation table used to translate characters sent from the host system to 3270 emulation. This parameter is valid only when KBDTYPE(*TRNTBL) is specified. Both TRNTBLOUT and TRNTBLIN must be specified if KBDTYPE(*TRNTBL) is specified.

***KBDTYPE:** Specifies that translation is done using the language specified in the KBDTYPE parameter.

The possible library values are:

***LIBL:** The library list is used to locate the outgoing translation table.

***CURLIB:** The current library for the job is used to locate the outgoing translation table. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the outgoing translation table is located.

user-defined-outgoing-table: Specify the qualified name of the table used for outgoing translation.

TRNTBLIN Parameter

Specifies the incoming translation table used to translate characters sent from 3270 emulation to the host system. This parameter is only valid when KBDTYPE(*TRNTBL) is specified. Both TRNTBLOUT and TRNTBLIN must be specified if KBDTYPE(*TRNTBL) is specified.

***KBDTYPE:** Specifies that translation is done using the language specified in the KBDTYPE parameter.

The possible library values are:

***LIBL:** The library list is used to locate the incoming translation table.

***CURLIB:** The current library for the job is used to locate the incoming translation table. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the incoming translation table is located.

user-defined-outgoing-table: Specify the qualified name of the table used for incoming translation.

Example

```
STREML3270   EMLCTL(TSOHOST)  PAGEUP(*F7)  PAGEDOWN(*F8)
```

This command places the display device into an emulation session that uses the first available device on the controller description TSOHOST for which the user has authority. When there are fewer input fields on the display than the maximum allowed by the 5250 and the Page Up key is pressed, an F7 key value is sent to the host system. When the Page Down key is pressed, an F8 key value is sent to the host system.

STRFMA (Start Font Management Aid) Command

The Start Font Management Aid (STRFMA) command shows the main Font Management Aid (FMA) menu. From this menu you can select the following options:

- Work with copy user-defined characters from work station font file to the double-byte character set (DBCS) font table.
- Copy user-defined characters from the double-byte character set font table to work station font file.
- Distribute dictionary or font file of the work station.

```
                                                            Required
STRFMA──────────────────────────────────────────────────────────────

                                                  ┌──────────────────┐
                                                  │Job: I  Pgm: I    │
                                                  └──────────────────┘
```

There are no parameters for this command.

Example

```
STRFMA
```

This command displays the main FMA menu.

STRIDD (Start Interactive Data Definition Utility) Command

The Start Interactive Data Definition Utility (STRIDD) command displays the main IDDU menu. From this menu, you can select an option that allows you to work with data definitions, data dictionaries, files, and libraries, or use related commands and office tasks.

```
                                                                Required
STRIDD────────────────────────────────────────────────────────────────────

                                                      ┌──────────────────────┐
                                                      │Job: I  Pgm: I  Exec  │
                                                      └──────────────────────┘
```

There are no parameters for this command.

Example

```
STRIDD
```

This commands displays the main IDDU menu.

STRIDXSCH (Start Index Search) Command

The Start Index Search (STRIDXSCH) command allows you to enter a command on any command line to access the system index, QHSS1, without having to press Help.

```
STRIDXSCH─────────────────────────────────────────────────────────────

                                                    Job: I  Pgm: I
```

There are no parameters for this command.

Example

```
STRIDXSCH
```

This command starts the Start Index Search function.

STRITF (Start Interactive Terminal Facility) Command

The Start Interactive Terminal Facility (STRITF) command allows the system user to send and receive data and file members for asynchronous display stations. The system user can also send documents. Before you can use Interactive Terminal Facility (ITF), you must vary on asynchronous communications descriptions.

```
                                                      Required
                                           ┌─┐
                                           │P│
STRITF──RMTLOCNAME remote-location-name────┴─┴──────────────────────
                                                 ┌─────────────────┐
                                                 │Job: I  Pgm: I  Exec│
                                                 └─────────────────┘
```

RMTLOCNAME Parameter
Specifies the name of the remote location you want to communicate with. This name is the same as the remote location name specified during configuration.

Example

```
STRITF  CHICAGO
```

The remote location name you wish to communicate with is CHICAGO.

STRJOBTRC (Start Job Trace) Command

The Start Job Trace (STRJOBTRC) command starts the OS/400 job tracing function to collect performance statistics for the specified job. It issues the following command:

```
TRCJOB MAXSTG(maxstg) TRCFULL(*STOPTRC)
```

The MAXSTG value used is either the STRJOBTRC default or specified value. The STRJOBTRC command issues a Service Job (SRVJOB) command if a job other than the current job is specified.

Once job tracing is started, a trace record is created for every external (program) call and return, exception, message, and work station wait in the job. At least two, and usually more, trace records are created for every I/O statement (open, close, read, write) in a high-level language program.

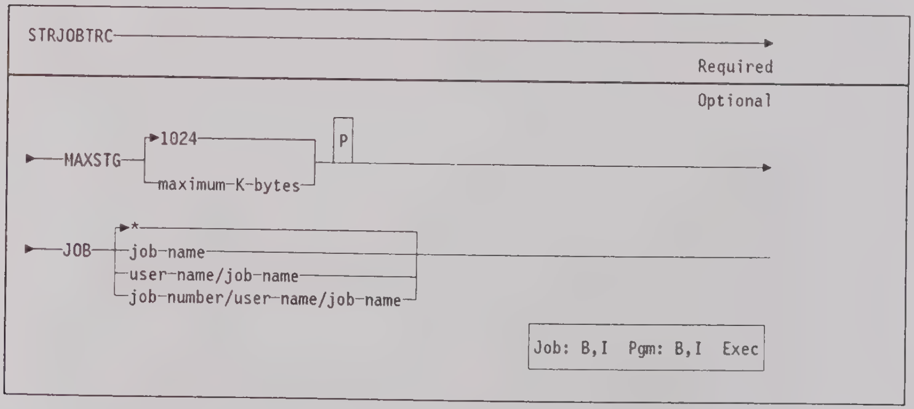
After the target programs have been run, the End Job Trace (ENDJOBTRC) command is used to turn tracing off, record the collected information in a data base file, and optionally produce reports used to analyze the data (the Print Job Trace (PRTJOBTRC) command may also be used to print the same report at any time thereafter). Alternatively, the command

```
TRCJOB SET(*CNL),
```

may be used to cancel the job trace (turn it off without recording any collected data).

Tracing has a significant impact on the performance of the current job. It also affects the performance of the system in general, but to a lesser extent.

The STRJOBTRC command is part of the IBM AS/400 Performance Tools Product, Program 5728-PT1. More information on the performance tools is in the *Performance Tools Guide.*



MAXSTG Parameter

Specifies the maximum amount of storage to allocate to the internal buffer where every trace record that is created is stored. If enough trace records are collected to fill the buffer, tracing is automatically turned off.

<u>1024</u>: A 1024K (1 megabyte) trace buffer is used. This size can handle about 14,000 trace records, which is sufficient in most cases.

maximum-K-bytes: Specify the size of the trace buffer in kilobytes. Each kilobyte can hold almost 14 trace records.

JOB Parameter

Specifies the job to trace.

<u>*</u>: Specifies the current job is to be traced.

job-name: Specify the name of the job that is traced. User name and job number qualifiers may be specified.

Example

```
STRJOBTRC   MAXSTG(512)
```

This command starts collecting trace data for the current job. It uses a 512K trace buffer, which is big enough to hold about 7000 trace records.

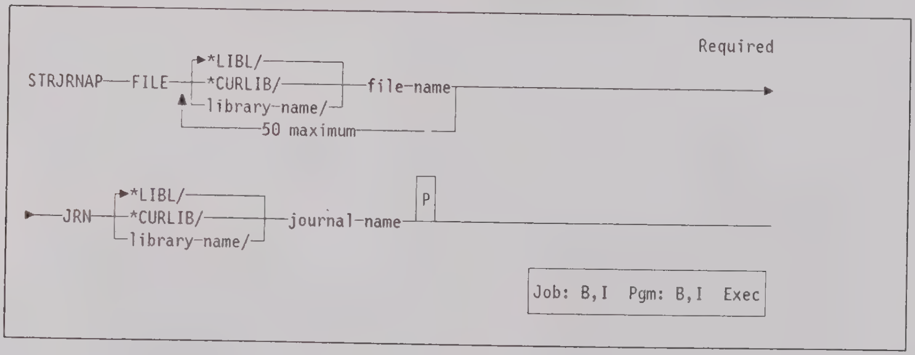
STRJRNAP (Start Journal Access Path) Command

The Start Journal Access Path (STRJRNAP) command is used to start journaling the access path, or access paths, for all members of a data base file to a specific journal. Any new member that is added to the file also has its access path journaled.

If a physical file is keyed, journaling can be started for its access paths. When access path journaling is started for a physical file, only the access paths for the physical file members are journaled. Journaling for any logical file access paths is started only when access path journaling is started for the logical file.

The journaled entries created after running this command cannot be used in any apply or remove journal changes operation. These entries are used only while loading the system after an abnormal system end to recover the access path without having to build it again in its entirety.

Restrictions: (1) Before journaling an access path, all physical files over which the access path is built must first be journaled to the same journal that is used to journal the access path. Even if all physical file members for a particular physical file are removed from the access path of a logical file, all physical files must still be journaled to the same journal before journaling the access path. (2) All access paths to be journaled must specify MAINT(*IMMED) or MAINT(*DLY) and FRCACCPTH(*NO). (3) If only after images are being journaled for the physical file members, the system automatically starts journaling the before and after images for the physical file once journaling is started for any access path built over the physical file. When journaling is ended for the access paths, the system automatically stops journaling the before images for the physical file and again only journals the after images. (4) Overrides are not applied to files listed in the FILE parameter.



FILE Parameter
Specifies the qualified name of the data base file that is to have its access paths journaled (library-name/file-name).

The possible library values are:

***LIBL:** The library list is used to locate the file.

***CURLIB:** The current library for the job is used to locate the file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the file is located.

JRN Parameter
Specifies the qualified name of the journal that is to receive the journaled access path changes (library-name/journal-name).

The possible library values are:

***LIBL:** The library list is used to locate the journal.

***CURLIB:** The current library for the job is used to locate the journal. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the journal is located.

Example

```
STRJRNAP FILE(MYFILE) JRN(MYLIB/JRNLA)
```

This command journals all access paths for all members in file MYFILE (as found using the library search list) to journal JRNLA in library MYLIB.

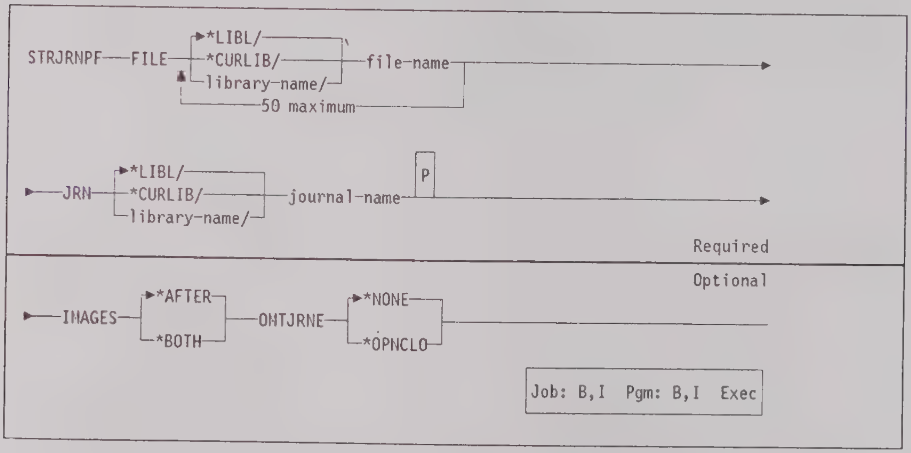
STRJRNPF (Start Journal Physical File) Command

The Start Journal Physical File (STRJRNPF) command is used to start journaling changes (made to all members of a specific physical file) into a specific journal. New members added to the file also have their changes journaled.

You can specify if you want only the *after* image or both *before* and *after* images of records in the physical file recorded in the journal. *Before* images are necessary for use by the IBM-supplied backout command, Remove Journaled Changes (RMVJRNCHG).

After journaling begins for the file, you should run the Save Changed Object (SAVCHGOBJ) command with OBJTYPE(*FILE) and OBJJRN(*YES) specified. The file must be saved because journaled changes cannot be applied to a version of the file that was saved when journaling was not in effect.

Restrictions: (1) The file must not be journaling changes to another journal. (2) There can be no members of the file currently in use for any reason on the system. (3) Overrides are not applied to files listed in the FILE parameter. (4) The maximum number of objects that can be associated with one journal is 262136. This maximum includes physical file members that are currently being journaled, members for which journaling was ended while the current receiver was attached, and journal receivers that are or were associated with the journal while the current journal receiver is attached. If the number of objects is larger than this maximum, journaling is not started.



FILE Parameter

Specifies the qualified name of the physical file that is to have its changes journaled (library-name file-name).

The possible library values are:

***LIBL:** The library list is used to locate the file.

***CURLIB:** The current library for the job is used to locate the file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the file is located.

JRN Parameter

Specifies the qualified name of the journal that receives the journaled changes (library-name/journal-name).

The possible library values are:

***LIBL:** The library list is used to locate the journal.

***CURLIB:** The current library for the job is used to locate the journal. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the journal is located.

IMAGES Parameter

Specifies the record images that will be written to the journal receiver for updates to records in the file.

***AFTER:** Only *after* images are generated for changes to records in this file.

***BOTH:** The system generates both *before* and *after* images for updates to records in this file.

OMTJRNE Parameter
Specifies the entries that are omitted.

*NONE: No entries are omitted.

***OPNCLO:** Open and close entries are omitted. Open and close operations on the specified file members will not generate open and close journal entries. (This prevents the use of TOJOBO and TOJOBC entries on the Apply Journal Changes (APYJRNCHG) and Remove Journal Changes (RMVJRNCHG) commands but saves some storage in the attached journal receivers.)

Example

```
STRJRNPF  FILE(MYFILE)  JRN(MYLIB/JRNLA)
```

This command journals all changes to all members of file MYFILE (as found using the library search list) to journal JRNLA in library MYLIB. Only the *after* images of updated records will be written to the journal.

STRKPF (Start Kanji Printer Function) Command

The Start Kanji Printer Function (STRKPF) command shows the main Kanji Printer Function (KPF) menu. From this menu you can select an option that allows you to create a form description, merge the user data and the form description, display the form description list, and load the font to the 5583 printer.

```
                                                               Required
STRKPF─────────────────────────────────────────────────────────────────

                                                         Job: I  Pgm: I
```

There are no parameters for this command.

Example

```
STRKPF
```

This command displays the main KPF menu.

STRMOD (Start Mode) Command

The Start Mode (STRMOD) command starts a single mode or all modes currently in use for an APPC remote location. The *APPC and APPN User's Guide* has more information on the STRMOD command.

```
STRMOD──RMTLOCNAME remote-location-name──────────────────────────────►
                                                             Required
─────────────────────────────────────────────────────────────────────
                                                             Optional
                                    ┌►*NETATR──┐
        ┌►*LOC──────────┐ ┌─┐       ├─*ALL─────┤
►──DEV──┤               ├─┤P├──MODE─┼─BLANK────┼─────────────────────►
        └─device-name───┘ └─┘       └─mode-name┘
                                                ┌►*LOC──────────────┐
                   ┌►*LOC─────────────┐         ├─*NETATR───────────┤
►──LCLLOCNAME──────┼─*NETATR──────────┼─RMTNETID┼─*NONE─────────────┼─────
                   └─local-location-name┘       └─remote-network-ID─┘
                                               ┌──────────────────────────┐
                                               │Job: B,I  Pgm: B,I   Exec │
                                               └──────────────────────────┘
```

RMTLOCNAME Parameter
Specifies the remote location name for which one or more modes is being started.

DEV Parameter
Specifies the device description used with the remote location.

***LOC:** The device associated with the remote location is used. If several devices can be associated with the remote location, the system determines which device is used.

device-name: Specify the name of the device being used.

MODE Parameter
Specifies the mode that is started.

***NETATR:** Specifies that the mode specified in the network attributes is used.

***ALL:** Specifies all modes currently in use by the remote location are being started.

BLANK: The mode name, consisting of 8 blank characters, is used.

mode-name: Specify the mode name being started.

LCLLOCNAME Parameter
Specifies the name of your location.

***LOC:** Specifies that the system selects the local location name. If several local location names are associated with the remote location, the system automatically selects the local location name.

***NETATR:** Specifies the local location name specified in the network attributes is used.

local-location-name: Specify the name of your location. The local location name is specified if you want to indicate a specific local location name to be associated with the remote location.

RMTNETID Parameter
Specifies the remote network ID used with the remote location.

***LOC:** Specifies that the system selects the remote network ID.

***NETATR:** Specifies that the local network ID specified in the network attributes is used.

***NONE:** The remote network ID is not specified.

remote-network-ID: Specify the name of the remote network ID used.

Example

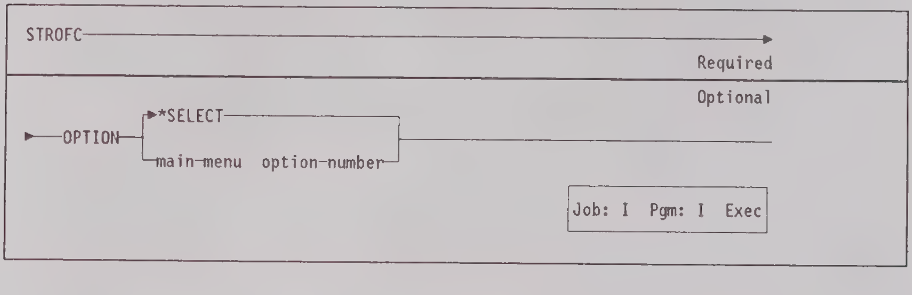
```
STRMOD  RMTLOCNAME(APPCRLOC)  DEV(APPCDEV)  MODE(APPCMODE)  RMTNETID(CHICAGO)
```

This command starts a mode named APPCMODE for a remote location named APPCRLOC, a device named APPCDEV, and a remote network ID of CHICAGO.

STROFC (Start Office) Command

The Start Office (STROFC) command allows you to use the office functions provided by the AS/400 Office.

Restriction: You must be enrolled in the system distribution directory and in AS/400 Office to use this command. You must also have AS/400 Office LPP installed.



OPTION Parameter
Specifies a number that corresponds to an option on the Office main menu.

SELECT: Displays the main menu for the office product. AS/400 Office attention key handling is called after any option is selected. Any attention key event defined before using AS/400 Office is suspended for the remainder of the session.

main-menu option-number: Specify a number, between 1 through 9, or 50, that corresponds to a main menu option. (If you select this parameter value, the main menu display is not shown; the display represented by the menu option number is diplayed instead.) AS/400 Office attention key handling is not called. Any attention key event defined before using AS/400 Office remains active.

Example

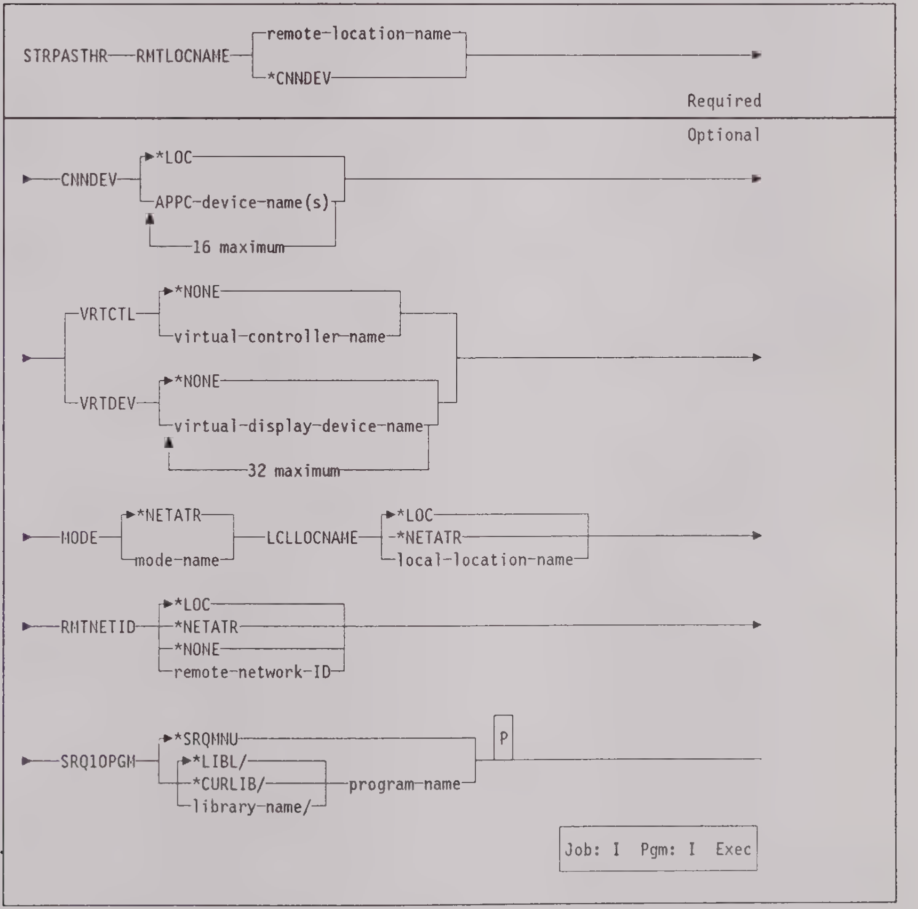
```
STROFC OPTION(*SELECT)
```

This command starts the AS/400 Office support and displays the Office main menu for the office product.

STRPASTHR (Start Pass-Through) Command

The Start Pass-Through (STRPASTHR) command allows you to pass-through to a remote system where you can sign on as if you were attached locally. This command uses the AS/400 advanced program-to-program communications (APPC) and advanced peer-to-peer networking (APPN) support to connect your display station to the remote system. Information on configuring or operating pass-through is in the *Communications User's Guide.*

Restrictions: (1) This command is not valid if entered in a job that is already in pass-through mode on a remote system. (2) This command cannot be entered at a work station with a screen of 12 lines by 80 characters.



RMTLOCNAME Parameter
Specifies the name of the remote location which is the target of the pass-through session.

remote-location-name: Specify the name of the target system in the APPN network.

***CNNDEV:** Specifies that the APPC devices specified by the CNNDEV parameter are used.

CNNDEV Parameter
Specifies the name(s) of the APPC device description(s) that connects the source system with the target system. If the target system is more than one system away and pass-through must establish the intermediate sessions, a list of nonnetworking APPC device descriptions must be specified. The nonnetworking APPC device names must be listed in the order that the systems are passed through to get to the target system.

__*LOC:__ Specifies that the RMTLOCNAME parameter is used to determine the target of the pass-through session and the path used to reach the target.

APPC-device-name(s): Specifies the names of the APPC device descriptions on the source system (and intermediate systems, if any), that connect the source system to the target system. Up to 16 names can be specified.

VRTCTL Parameter

Specifies the name of the virtual controller on the remote system that is used to do pass-through. If you specify a virtual controller, one of the virtual display devices attached to it is selected for your pass-through job. The system tries to compare the device type and model of your physical display device with an available virtual device. If the same type or a similar type is not available, a comparison is made with a 5251 Model 11 (if available). If a 5292 Model 2 device is not available, but a 5251 Model 11 is, graphics cannot be done. If no virtual devices are available, you must try another virtual controller or try again later.

This parameter is mutually exclusive with the VRTDEV parameter; *NONE must be specified when VRTDEV is specified with a list of devices.

***NONE:** No controller is specified.

virtual-controller-name: The name of the virtual controller description on the remote system.

VRTDEV Parameter

Specifies one or more devices on the remote system that are connected to a virtual controller used for the pass-through session. A device on the remote system from the list is selected based on a comparison of device type and model. If more than one device is in the list, the first available device that closely matches the type and model of the device on which the command is entered is used. If the same type or a similar type is not available, a comparison is made with a 5251 Model 11 (if available). If a 5292 Model 2 device is not available, but a 5251 Model 11 is, graphics cannot be done. This parameter is mutually exclusive with the VRTCTL parameter; *NONE must be specified when VRTCTL is specified with a controller name.

***NONE:** No device names are specified.

virtual-display-device-name: Specify the name(s) of the virtual display device description on the remote system. Up to 32 names can be specified.

MODE Parameter

Specifies the mode name used for pass-through. This parameter is not valid if RMTLOCNAME(*CNNDEV) is specified. In this case, the system uses the first mode name specified in the APPC device description.

***NETATR:** Specifies that the mode in the network attributes is used.

mode-name: Specify a mode name. Specify BLANK for a mode name consisting of 8 blank characters.

LCLLOCNAME Parameter

Specifies the local location name for the source system.

***LOC:** Specifies that the local location name is chosen by the system.

***NETATR:** Specifies that the default location name defined in the Change Network Attributes (CHGNETA) command is used.

local-location-name: Specify the local location name for the source system.

RMTNETID Parameter

Specifies the network identifier of the network where the remote location resides. This parameter is not valid if RMTLOCNAME(*CNNDEV) is specified.

***LOC:** Specifies that the system selects the remote network ID.

***NETATR:** Specifies that the local network ID specified in the network attributes is used.

***NONE:** Specifies that the remote location does not support network ID.

remote-network-ID: Specify the identifier of the network where the target system (specified by the RMTLOCNAME parameter) is located.

SRQ10PGM Parameter

Specifies the System Request menu or a user-written program starts when SYSREQ option 10 is selected. The user program can display a menu that allows you to select the system you want to access, and then transfer to a group job that sends the STRPASTHR command to the desired system. More information is in the *Communications User's Guide.*

***SRQMNU:** The System Request menu is displayed.

The possible library values are:

***LIBL:** The library list is used to locate the program.

***CURLIB:** The current library for the job is used to locate the program. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the program is located.

program-name: Specify the qualified name of the program started when SYSREQ option 10 is selected.

Example

```
STRPASTHR  RMTLOCNAME(*CNNDEV) CNNDEV(DET CHI TOR) VRTCTL(VWSC)
```

This command specifies that you want to pass-through to the Toronto system and go through Detroit and Chicago. The *Communications User's Guide* has more information on how this works.

STRPCCMD (Start PC Command) Command

The Start Personal Computer Command (STRPCCMD) command allows you to run a single application or disk operating system (DOS) command on an attached personal computer.

```
STRPCCMD──PCCMD PC-command──────────────────────────────────────────►
                                                            Required
─────────────────────────────────────────────────────────────────────
                                                            Optional
              ┌─*YES─┐
►──PAUSE──────┤      ├──────────────────────────────────────────────
              └─*NO──┘

                                           ┌─────────────────────────┐
                                           │Job: B,I  Pgm: B,I  Exec │
                                           └─────────────────────────┘
```

PCCMD Parameter
Specifies the valid DOS command or personal computer application you want to run. If the command or application name contains blanks or commas, you must enclose the command in apostrophes.

PAUSE Parameter
Specifies if the computer should pause after running a command.

***YES:** The computer must pause after running the command before returning to the AS/400 PC Support

***NO:** The computer returns immediately to the AS/400 PC Support menu after the command runs.

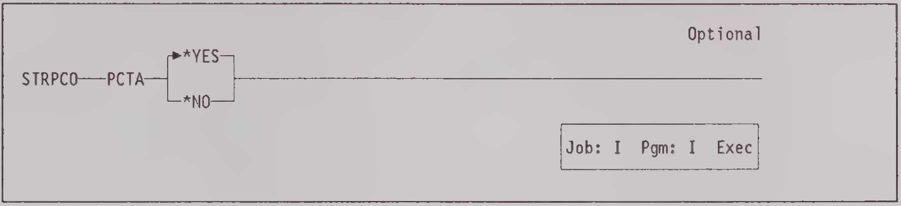
Example

```
STRPCCMD  'DIR\W'  PAUSE(*NO)
```

This command starts the DOS command DIR\W. After the command runs, the personal computer immediately returns you to the AS/400 PC Support menu.

STRPCO (Start PC Support Organizer) Command

The Start PC Support Organizer (STRPCO) command starts the PC Support organizer on the host system.



PCTA Parameter
Specifies if you want to use the Personal Computer Text Assist (PCTA).

<u>*YES:</u> The Personal Computer Text Assist is used.

***NO:** The Personal Computer Text Assist is not used.

Example

```
STRPCO
```

This command starts the AS/400 PC Support organizer on the host system.

STRPDM (Start Programming Development Manager) Command

The Start Programming Development Manager (STRPDM) command calls the Programming Development Manager (PDM) utility. A menu is displayed allowing you to choose selections so you can work with libraries, objects, members, and user-defined options.

```
STRPDM──────────────────────────────────────────────────────────────────

                                                  ┌────────────────────┐
                                                  │Job: I  Pgm: I  Exec│
                                                  └────────────────────┘
```

There are no parameters for this command.

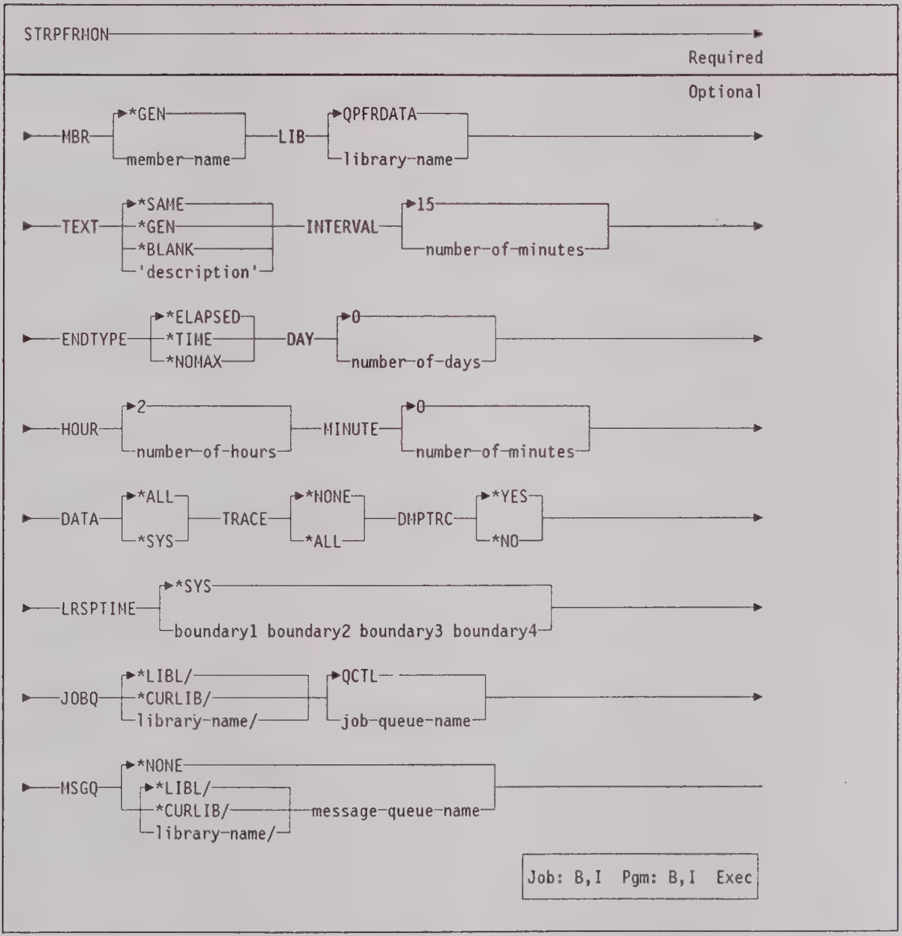
Example

```
STRPDM
```

This command calls the PDM utility.

STRPFRMON (Start Performance Monitor) Command

The Start Performance Monitor (STRPFRMON) command submits the performance monitoring job (QPFRMON) to the job queue identified in the JOBQ parameter. Once the job is started, it periodically collects performance data and puts it into system-supplied data base files. The data is collected as often as specified in the INTERVAL parameter.



MBR Parameter

Specifies the data base file member where the output for the collected data is directed.

__*GEN:__ Specifies that the member name is created by the system. The name of the member that is created is Qyydddhhmm, where yy is the year, ddd is the date in Julian format, and hhmm is the time (hours and minutes) that the performance monitor job was started.

member-name: The name of the member that is used by the system. If a member does not exist, the system creates a member by the specified name.

LIB Parameter

Specifies the library where the data base files for performance data is collected. Each file that is not found in the specified library is automatically created by the system (in the library).

__QPFRDATA:__ Specifies that the IBM-supplied performance library, QPFRDATA, is used to locate the data base files.

library-name: Specify the name of the library where the data base files reside.

TEXT Parameter

Specifies text that briefly describes the data base member that receives the performance data. This text is on each member, within the multiple files, associated with a measurement period.

***SAME:** If the member already exists, no change is made to the descriptive text. If the member is created, no text is specified.

***GEN:** If the member is created by the system, the text is 'Performance data collected dddddd at hhmm', where dddddd is the system date (in the format specified by system value (QDATFMT), and hhmm is the current time in hours and minutes. If the member already exists, no change is made.

***BLANK:** No text is specified.

'description': Specify a description of no more than 50 characters, enclosed in apostrophes.

INTERVAL Parameter
Specifies the time interval (in minutes) between each collection of system performance data.

15: The default collection interval value of 15 minutes is used.

number-of-minutes: Specify a collection interval value between 5 and 60 minutes.

ENDTYPE Parameter
Specifies how the performance monitor determines when to stop collecting data.

***ELAPSED:** Collection stops after the specified amount of time has elapsed (as specified in the hour and minute parameters). Changes to the system clock do not affect the stop time (once the performance monitor has been started).

***TIME:** Collection stops at the specified time on the specified day (as specified in the day, hour, and minute parameters).

***NOMAX:** Collection does not stop until either the End Performance Monitor (ENDPFRMON) command is issued, or any of the data base files where the collected data is sent fills up.

The trace data collection duration is limited due to the internal trace table size. Its duration is also dependent on the level of activity taking place on the system. If trace is being requested (TRACE(*ALL)), the total data collection period should not exceed 12 hours. The trace function will stop when the trace table is full. The monitor will continue to collect interval data.

DAY Parameter
Specifies the number of days to collect data when ENDTYPE(*ELASPED) is specified. Specifies the number of days from the current day that collection is to end when ENDTYPE(*TIME) is specified. A value of 0 to 7 may be specified. If ENDTYPE(*NOMAX) is specified, this parameter has no meaning and is ignored.

0: The default value is 0 days.

number-of-days: If ENDTYPE(*ELAPSED) is specified, specify the number of days to collect data. When ENDTYPE(*TIME) is specified, specify the number of days from the current day that data collection is to end. A value from 0 to 7 may be specified.

HOUR Parameter
Specifies the number of hours to collect data when ENDTYPE(*ELAPSED) is specified. Specifies the hour of the day that data collection ends when ENDTYPE(*TIME) is specified.

If ENDTYPE(*NOMAX) is specified, this parameter has no meaning and is ignored.

2: The default value is 2 hours.

number-of-hours: If ENDTYPE(*ELAPSED) is specified, specify the number of hours to collect data. A value from 0 to 999 may be specified. If ENDTYPE(*TIME) is specified, specify the hour of the day that data collection ends. A value from 0 to 23 may be specified.

MINUTE Parameter
Specifies the number of minutes to collect data when ENDTYPE(*ELAPSED) is specified.

Specifies the minute of the specified hour that data collection ends when ENDTYPE(*TIME) is specified.

If ENDTYPE(*NOMAX) is specified, this parameter has no meaning and is ignored.

0: The default value is 0 minutes.

number-of-minutes: If ENDTYPE(*ELAPSED) is specified, specify the number of minutes to collect data. A value from 0 to 99 may be specified.

If ENDTYPE(*TIME) is specified, specify the minute of the hour that data collection is to end. A value from 0 to 59 may be specified.

DATA Parameter
Specifies the type of data collected.

***ALL:** Specifies that all data (system data and communications data) is collected.

***SYS:** Specifies that only system data is collected.

TRACE Parameter
Specifies what type of internal trace is started.

***NONE:** Specifies that no trace is started.

***ALL:** Specifies that all of the internal traces that contain performance related information are started.

DMPTRC Parameter
Specifies whether the trace is dumped when the data collection ends.

***YES:** The trace is dumped when the data collection ends.

***NO:** The trace is not dumped when the data collection ends. It may be dumped at a later time via the Dump Trace (DMPTRC) command.

LRSPTIME Parameter

Specifies the local work station response time categories. The performance monitor keeps track of interactive response times for each local work station attached to a controller that supports collecting response time data. The response times are grouped into five categories and this parameter allows definition of the categories. The values specified must be a three-position decimal number, with one decimal position.

***SYS:** The default categories are: 0-1 seconds, 1-2 seconds, 2-4 seconds, 4-8 seconds, and greater than 8 seconds.

boundary1: First response time boundary. All responses falling between zero and this boundary value are counted in the first response time category. This is a DEC(3,1) variable.

boundary2: Second response time boundary. All responses falling between the first boundary value and this boundary value are counted in the second response time category. This is a DEC(3,1) variable.

boundary3: Third response time boundary. All responses falling between the second boundary value and this boundary value are counted in the third response time category. This is a DEC(3,1) variable.

boundary4: Fourth response time boundary. All responses falling between the third boundary value and this boundary value are counted in the fourth response time category. This is a DEC(3,1) variable.

All response times that are greater than this value are counted in the fifth response time category.

JOBQ Parameter

Specifies the job queue where the performance monitor job is submitted. The job is started by the subsystem monitor that allocates the job queue.

The possible library values are:

***LIBL:** The library list is used to locate the job queue.

***CURLIB:** The current library for the job is used to locate the job queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the job queue is located.

QCTL: The IBM-supplied controlling subsystem, QCTL, is used.

job-queue-name: Specify the name of an alternative job queue.

MSGQ Parameter

Specifies the message queue where the performance monitor sends messages, in addition to the system operator's message queue (QSYS/QSYSOPR). Messages sent to the message queue communicate status information relating to performance monitor startup, end, and problems.

***NONE:** Messages are sent to the system operator message queue only.

The possible library values are:

***LIBL:** The library list is used to locate the message queue.

***CURLIB:** The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message queue is located.

message-queue-name: Specify the qualified name of the message queue where messages are sent.

Examples

```
STRPFRMON INTERVAL(30) ENDTYPE(*TIME) DAY(7) HOUR(16) MINUTE(30)
```

In this example, the performance monitor is submitted to the QCTL job queue in the QSYS library. The performance monitor generates the member name (based upon the time that the performance monitor is started from the job queue), it puts performance data into the performance data base files in library QPFRDATA every 30 minutes, and ends 7 days from the current day at 4:30 in the afternoon.

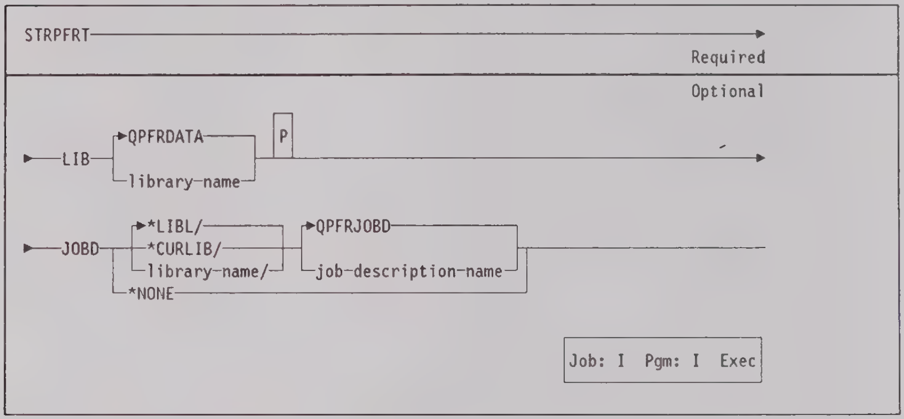
```
STRPFRMON MBR(JULY12) ENDTYPE(*NOMAX) JOBQ(QGPL/QBATCH) MSGQ(MYMSGQ)
```

In this example, the performance monitor is submitted to the QBATCH job queue in the library QGPL. Performance data is collected in the performance data base files QPFRDATA and the member name is JULY12. The performance does not have a time at which it will end (it may be ended by entering the End Performance Monitor (ENDPFRMON) command). Messages indicating the status of the performance monitor is sent to the MYMSGQ message queue (in addition to the QSYSOPR message queue).

STRPFRT (Start Performance Tools) Command

The Start Performance Tools (STRPFRT) command calls the Performance Tools menu interface. This shows the Performance Tools main menu (PERFORM), where you have access to all of the performance tools.

The STRPFRT command is part of the IBM AS/400 Performance Tools product, Program 5728-PT1. More information on the Performance Tools program is in the *Performance Tools Guide*.



LIB Parameter
Specifies the library where the data base files for performance data reside.

QPFRDATA: The IBM-supplied performance data library, QPFRDATA, is where the data base files resides

library-name: Specify the name of the library where the data base files is located.

JOBD Parameter
Specifies the job description used by the data reduction jobs for batch processing.

The possible library values are:

***LIBL:** The library list is used to locate the job description.

***CURLIB:** The current library for the job is used to locate the job description. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the job description is located.

QPFRJOBD: Specifies the IBM-supplied performance tools job description.

job-description-name: Specify the name of an alternate job description.

***NONE:** A batch job is not submitted; instead, processing continues interactively while the user waits. The user's work station is not available for other use during this time, which could be significant for long jobs.

Example

```
STRPFRT
```

This command calls the menu interface to the performance tools. The library, QPFRDATA, is used in storing and retrieving the performance data for this session. Any requests that are submitted to run in batch will use the job description QPFRJOBD in library QPFR.

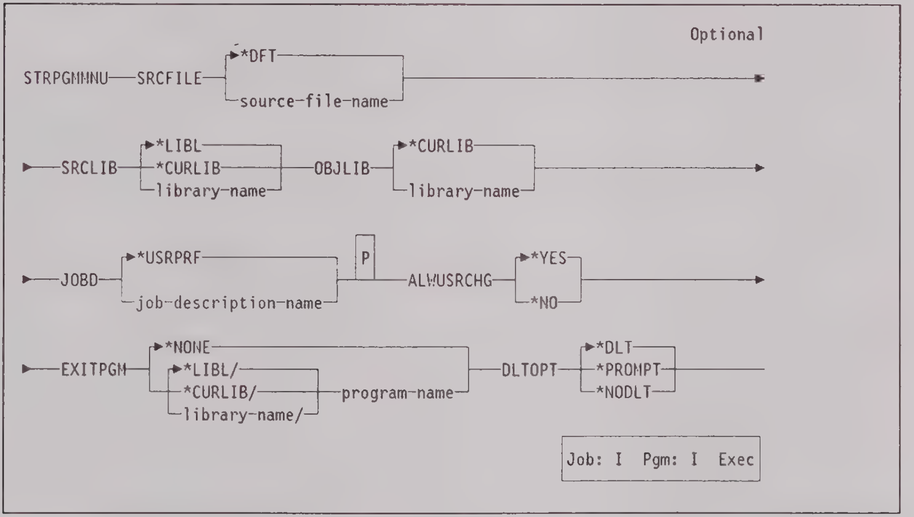
STRPGMMNU (Start Programmer Menu) Command

The Start Programmer Menu (STRPGMMNU) command shows the Programmer Menu. This command can be used instead of the CALL QPGMMENU function, and allows you to pass parameters to specify and control the data which appears in the associated fields on the programmer menu.

Notes:

1. A user exit program can be called instead of submitting a job when option 3 is selected.
2. The first four parameters control the defaults that appear when the menu is first displayed.

The *CL Programmer's Guide* has more information about using the Programmer Menu.



SRCFILE Parameter
Specifies the name of an existing source file that is to be used on the Programmer's Menu.

***DFT:** The source file will default to the default for the TYPE of source file being specified on the menu. This field is blank when shown on the display station.

file-name: Specify the name of the source file to be used on the Programmer's Menu.

SRCLIB Parameter
Specifies the qualified name of the library that is searched when looking for the source file.

The possible library values are:

***LIBL:** The library list is used to locate the source file.

***CURLIB:** The current library for the job is used to locate the source file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the source file is located.

OBJLIB Parameter
Specifies the name of the library which was created on the Programmer's Menu, receives objects.

***CURLIB:** The current library for the job is the library containing or receiving the object.

library-name: Specify the name of the library which contains or receives the created object.

JOBD Parameter
Specifies the name of the job description used with the job being submitted.

***USRPRF:** The job description included in your user profile is used for the job.

job-description-name: Specify the name of the job description to use for the job. (The job description is found through the library list being used by the job.)

ALWUSRCHG Parameter
Specifies whether the menu screen fields you specified in the previous parameters in this command are input capable.

***YES:** Allows you to change the values on the display.

***NO:** The display fields are not input capable. (This option is intended to minimize errors, and should not be considered a security function.)

EXITPGM Parameter
Specifies the qualified name of a user-written program that is called as an exit program in place of submitting a batch job when menu option 3 is selected. When the exit program is called, parameters are sent by the Programmer Menu that must be received by the exit program. (Refer to the examples at the end of this command description.) The *CL Programmer's Guide* has more information about the EXITPGM parameter.

***NONE:** No user-written program is called; a batch job is submitted. When *NONE is specified, DLTOPT must be *DLT.

The possible library values are:

***LIBL:** The library list is used to locate the program.

***CURLIB:** The current library for the job is used to locate the program. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the program is located.

program-name: Specify the qualified name of the program called, when option 3 is selected, instead of submitting the create command as a batch job. When a value is specified for this parameter, the text that appears on the menu for option 3 shows the name and library of the exit program.

DLTOPT Parameter
Specifies the action to be taken when: (1) A program name is specified for the EXITPGM parameter; (2) option 3 of the Programmer Menu is selected; and (3) an object of the name and type to be created exists in the library specified by the OBJLIB field of the menu. Regardless of the value specified, the system passes a parameter (from among the parameters passed from the Programmer Menu) to your exit program that specifies whether or not the object exists.

***DLT:** This value must be specified if *NONE is specified for the EXITPGM parameter. If an EXITPGM is specified, and the object you specified to be created with option 3 exists and the Enter key is pressed, a message is shown; you must press F11 to proceed. If F11 is pressed, the system deletes or replaces the object before calling the program specified by the exit program. (This is the normal Programmer Menu function when an EXITPGM is not specified.)

Notes:

1. When this value is specified, the object does not exist when the job is submitted or when the user exit program is called.
2. The object is replaced rather than deleted if the system CRTxxx commmand is used.

***PROMPT:** The system does not delete or replace the object, but you are prompted for approval to delete the object. If the object exists, and the Enter key is pressed, a message is displayed. You must press F11 to proceed; the system does not delete the object. This allows you to confirm whether you want the object deleted or replaced, yet still allows the deletion to be controlled by the exit program.

***NODLT:** The user exit program is called regardless of the presence of the object.

The following chart shows the actions taken by the system when the exit program is called. The value passed to the exit program is a character variable of length 1 that contains the character value shown. Note that a value other than 0 indicates that the object does not exist.

DLTOBJ Specified	F11 Required[1]	Object Deleted[1]	Value Passed to Exit Program[1]	F11 Required[2]	Value Passed to Exit Program[2]
*DLT	Yes	Yes	1	No	2
*PROMPT	Yes	No	0	No	2
*NODLT	No	No	0	No	2

[1] Object did exist when option 3 was selected.
[2] Object did not exist when option 3 was selected.

Examples

```
STRPGMMNU
```

This command displays the Programmer Menu with defaults for all parameters. This has the same result as entering CALL QPGMMENU.

```
STRPGMMNU   SRCFILE(YOURFILE)  SRCLIB(YOURLIB)  +
  OBJLIB(YOURLIB)  JOB(YOURJOBD)  ALWUSRCHG(*NO)
```

This command prevents the values on the menu from being changed from those specified on the command.

```
STRPGMMNU   EXITPGM(OPT3PGM)  DLTOPT(*PROMPT)
```

This command calls user exit program OPT3PGM instead of submitting a batch job, when option 3 is specified. If the object already exists, DLTOPT(*PROMPT) requires you to use F11; however, the object is not deleted.

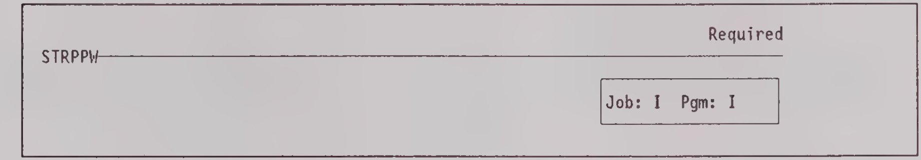
The following portion of a CL program is an example of how these parameters would be received by a user exit program:

```
PGM PARM(&OPTION &PARM &TYPE &PARM2  +
  &SRCFILE &SRCLIB  +
  &OBJLIB &JOBD  &RQSLEN  &RQSDTA512  +
  &F4  &F11 &EXIST)

/* The following values are passed in exactly as they appear  */
   on the Programmer Menu.                                    */
    DCL VAR(&OPTION)  TYPE(*CHAR)  LEN(2)  /*                 */
    DCL VAR(&PARM)    TYPE(*CHAR)  LEN(10) /*                 */
    DCL VAR(&TYPE)    TYPE(*CHAR)  LEN(4)  /*                 */
    DCL VAR(&PARM2)   TYPE(*CHAR)  LEN(21) /*                 */
    DCL VAR(&SRCFILE) TYPE(*CHAR)  LEN(10) /*                 */
    DCL VAR(&SRCLIB)  TYPE(*CHAR)  LEN(10) /*                 */
    DCL VAR(&OBJLIB)  TYPE(*CHAR)  LEN(10) /*                 */
    DCL VAR(&JOBD)    TYPE(*CHAR)  LEN(10) /*                 */

/* The following values are derived by QPGMMENU from the      */
/* information entered to the above fields and the F keys.   */

   DCL  VAR(&RQSLEN)  TYPE(*DEC)   LEN(3 0)/* NUMBER OF BYTES   +
                                          OF REQUEST DATA */
   DCL  VAR(&RQSDTA512)  TYPE(*CHAR)   LEN(512)/* DATA FOR
                  RRQSDTA PARAMETER OF SBMJOB COMMAND. */
   DCL  VAR(&F4)    TYPE(*CHAR)   LEN(1) /* F4 WAS PRESSED,   +
                                    '1', OTHERWISE '0'.   */
   DCL  VAR(&F11)   TYPE(*CHAR)   LEN(1)   /* F11 WAS PRESSED,+
                                   '1', OTHERWISE '0'.     */
   DCL  VAR(&EXIST)  TYPE(*CHAR)   LEN(1)  /* OBJECT EXISTS- '0'+
                                 OBJECT WAS DELETED- '1'      +
                                 OR OBJECT DID NOT EXIST -'2'*/
```

The *CL Programmer's Guide* has additional information, along with examples of the STRPGMMNU command with the EXITPGM parameter.

STRPPW (Start Page Printer Writer) Command

The Start Page Printer Writer (STRPPW) command shows the main Advanced Page Printer Writer (APPW) menu. From this menu you can select an option that allows you to create a form description, merge the user data and the form description, display the form description list, display the symbol menu, and load the font to the 5337 printer.



There are no parameters for this command.

Example

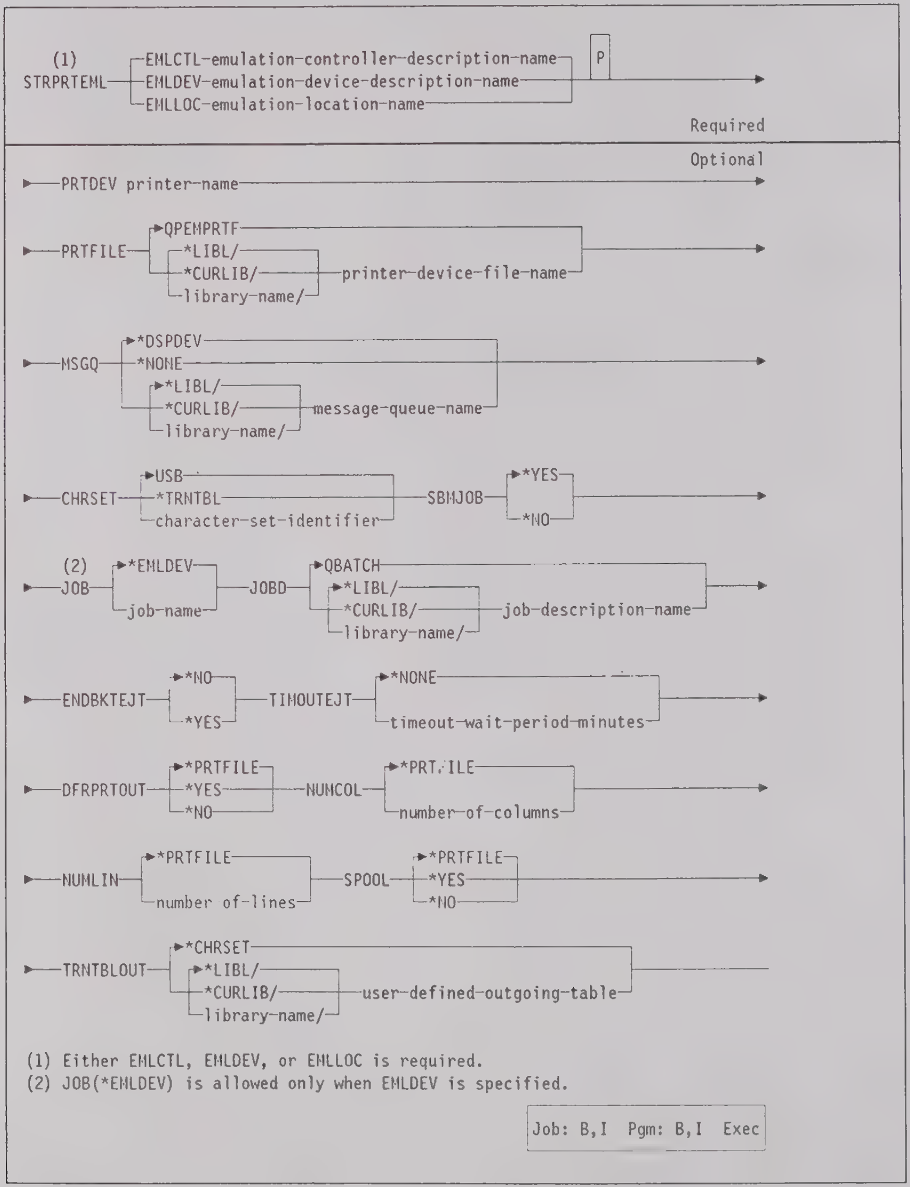
```
STRPPW
```

This command displays the main APPW menu.

STRPRTEML (Start Printer Emulation) Command

The Start Printer Emulation (STRPRTEML) command starts 3270 printer emulation using a binary synchronous communications (BSC) or systems network architecture (SNA) emulation printer device and a printer device file.

For more information, refer to the *3270 Device Emulation User's Guide.*



(1) Either EMLCTL, EMLDEV, or EMLLOC is required.
(2) JOB(*EMLDEV) is allowed only when EMLDEV is specified.

Job: B,I Pgm: B,I Exec

EMLCTL Parameter
Specifies the name of a BSC controller description (APPTYPE(*EML)) or SNA controller description type of HOST that has attached 3270 emulation printer device descriptions. When this parameter is specified, the printer emulation job uses a 3270 printer device attached to this controller description to which the requester is authorized and that is available.

EMLDEV Parameter
Specifies the name of a BSC or an SNA printer emulation device (EMLDEV(3284, 3286, 3287, 3288, or 3289)) that is used by the printer emulation job to do 3270 printer emulation. You must be

authorized to the device and the device must be available.

EMLLOC Parameter

Specifies the location name associated with this session. This name is defined during configuration and refers to the remote location where communication takes place.

PRTDEV Parameter

Specifies the name of the printer used with this printer device file to create the printed output. If the printer output for this printer is not being spooled, the printer must not be in use.

PRTFILE Parameter

Specifies the printer device file that is to print data received from the host system. The printer device file can be spooled or not spooled. If you want the files from the host system to be printed on the printer device, you must specify the printer device name in the printer file for files with SPOOL(*NO) specified. For files with SPOOL(*YES) specified, you must start a writer to the output queue for this job, and print to the associated printer device.

QPEMPRTF: The standard printer file (which specifies SPOOL(*YES)) shipped with the emulation utility is used as the printer device file.

The possible library values are:

***LIBL:** The library list is used to locate the printer device file.

***CURLIB:** The current library for the job is used to locate the printer device file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the printer device file is located.

printer-device-file-name: Specify the qualified name of a user-defined printer device file.

MSGQ Parameter

Specifies the message queue, if any, to which messages will be sent when the printer emulation job is running.

***DSPDEV:** The current display device message queue is used. If this value is specified from a batch job, no messages is sent to message queues other than the job log for the printer emulation job.

***NONE:** No messages are sent to message queues other than the job log for the printer emulation job.

The possible library values are:

***LIBL:** The library list is used to locate the message queue.

***CURLIB:** The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message queue is located.

message-queue-name: Specify the qualified name of the message queue where messages are sent when the printer emulation job is running.

CHRSET Parameter

Specifies the character set used by the printer that is being emulated.

USB: The printer being emulated uses the United States basic character set.

***TRNTBL:** Allows user-defined translate tables to be used. The character translation is defined in the translation table specified by the TRNTBLOUT parameter.

character-set-identifier: Specify the identifier of the character set used by the printer being emulated.

Character Translation Table

Language/Country	Identifier
Arabic X/Basic	CLB
Austria/Germany	AGB
Austria/Germany Multinational	AGI
Belgium Multinational	BLI
Canadian French	CAB
Canadian French Multinational	CAI
Cyrillic	CYB
Denmark	DMB
Denmark Multinational	DMI
Finland/Sweden	FNB
Finland/Sweden Multinational	FNI
France (Azerty)	FAB
France (Azerty) Multinational	FAI
France (Qwerty)	FQB
France (Qwerty) Multinational	FQI
Greece	GKB
Hebrew	NCB
Iceland	ICB
Iceland Multinational	ICI
International	INB
International Multinational	INI
Italy	ITB
Italy Multinational	ITI
Japan English	JEB
Japan English Multinational	JEI
Japan Kanji	JKB (For PS/55 and 5295 display stations)
Japan Katakana	KAB (For 5251, 5291, 5292, and 3180 Katakana display stations)
Korea	KOB
Netherlands	NEB
Netherlands Multinational	NEI
Norway	NWB
Norway Multinational	NWI
Simplified Chinese	RCB
Portugal	PRB

Portugal Multinational	PRI
Latin 2	ROB
Spain	SPB
Spain Multinational	SPI
Spanish Speaking	SSB
Spanish Speaking Multinational	SSI
Sweden	SWB
Sweden Multinational	SWI
Switzerland/French Multinational	SFI
Switzerland/German Multinational	SGI
Traditional Chinese	TAB
Thailand	THB
Turkey	TKB
United Kingdom	UKB
United Kingdom Multinational	UKI
United States/Canada	USB
United States/Canada Multinational	USI
Yugoslavia	YGI

Note: For example, CHRSET(USB) indicates the basic United States character set is used.

SBMJOB Parameter

Specifies if the printer emulation should be done as a separate job or as part of this job.

*YES: A specific job is submitted to do the printer emulation. The job attributes and job queue are determined from the job description (JOBD parameter). The job uses your user profile.

*NO: Printer emulation is done in the current job. This allows you to specify an Override Printer File (OVRPRTF) command before the printer emulation request to override the printer file. You can also follow the printer emulation request with any other system request, such as copy spooled file to data base file. SBMJOB(*NO) should be used only in a batch job; however, it is allowed in an interactive job.

JOB Parameter

Specifies a job name for the printer emulation job. If EMLDEV is not specified, and a batch job is to be submitted (SMBJOB(*YES)), a job name must be specified. This parameter is ignored for SBMJOB(*NO).

*EMLDEV: The job name is to be the same as the printer device name.

job-name: Specify the name to be assigned to the job.

JOBD Parameter

Specifies the job description for the job that will be submitted for 3270 printer emulation. This parameter is ignored for SBMJOB(*NO).

QBATCH: The job description QBATCH is to be used for the job.

The possible library values are:

*LIBL: The library list is used to locate the job description name.

*CURLIB: The current library for the job is used to locate the job description name. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the job description name is located.

job-description-name: Specify the qualified name of the job description associated with the job.

ENDBKTEJT Parameter

Specifies if SNA printer emulation should force out the emulation output when an SNA End Bracket (EB) is received from the host system. Emulation output is forced out by closing and then reopening the emulation printer file specified in the PRTFILE parameter. This parameter uses the default value of *NO for BSC printer emulation.

*NO: The emulation output is not forced out when SNA printer emulation receives an End Bracket.

*YES: The emulation output is forced out when SNA printer emulation receives an End Bracket. This is done only if the open printer file contains host system data.

TIMOUTEJT Parameter

Specifies if SNA printer emulation should force out the emulation output when a timeout has occurred while waiting for host system data. Emulation output is forced out by closing and then reopening the emulation printer file specified in the PRTFILE parameter. This parameter uses the default value of *NONE for BSC printer emulation.

*NONE: The emulation output is not forced out based on a specified timeout period.

timeout-wait-period-minutes: Specify a timeout wait period in minutes. When the SNA host system does not send data within this timeout period, the emulation output is forced out. The emulation output is forced out only if the open printer file contains host system data.

DFRPRTOUT Parameter

Specifies if spooled output should be printed immediately or should be delayed.

*PRTFILE: Specifies if the spooled output is printed.

*YES: Printing of spooled output is delayed until the printer is signed off of device emulation or a move is done.

*NO: Spooled output can be printed before the spool file is closed. The printed output does not contain all the data sent by the host system until the spool file is closed. If the printer is not using spooling, this parameter is ignored.

Note: Once the printer starts printing output from 3270 device emulation, spooled output from other jobs sharing the printer does not print until the spooled file that is currently printing is complete.

NUMCOL Parameter

Specifies the number of columns in a line when creating printed output.

***PRTFILE:** Specifies that the print file contains the starting number of columns per line.

number-of-columns: Specify the number of columns per line in the printed output. Valid values are 1 through 132.

NUMLIN Parameter

Specifies the maximum number of lines per page when creating the printed output.

***PRTFILE:** Specifies that the print file contains the number of lines per page.

number-of lines: Specify the number of lines per page in printed output. Valid values are 1 through 112.

SPOOL Parameter

Specifies if the output data for the printer device file is spooled.

***PRTFILE:** Specifies that the print file selects if spooling is done.

***YES:** The data is spooled.

***NO:** The data is not spooled; it is sent directly to the device and is printed as the output becomes available.

TRNTBLOUT Parameter

Specifies the outgoing translation table used to translate characters sent from the host system to the remote 3270 emulation. This parameter is valid only when CHRSET(*TRNTBL) is specified.

***CHRSET:** Specifies that translation is done when data is sent from the host system using the character set specified in the CHRSET parameter.

The possible library values are:

***LIBL:** The library list is used to locate the translation table.

***CURLIB:** The current library for the job is used to locate the translation table. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the translation table is located.

user-defined-outgoing-table: Specify the qualified name of the table which is used for outgoing translation.

Examples

```
STRPRTEML   EMLDEV(HOSTPRT4)
```

This command starts a batch job by accepting data from the HOSTPRT4 device and prints the data to the standard emulation printer file (QPEMPRTF). The job is named HOSTPRT4 and runs until the job is canceled. Messages are sent to the current work station message queue.

```
STRPRTEML   EMLDEV(HOSTPRT5)  SBMJOB(*NO)
```

This command does printer emulation in the current job by accepting data from the HOSTPRT5 device, and writing the data to the standard emulation printer device file (QPEMPRTF). The request is active until it ends through the End Printer Emulation (ENDPRTEML) command, or until the job is canceled.

```
STRPRTEML  EMLCTL(EMLCTL1)  DFRPRTOUT(*NO)  NUMLIN(96)  TIMOUTEJT(10)
```

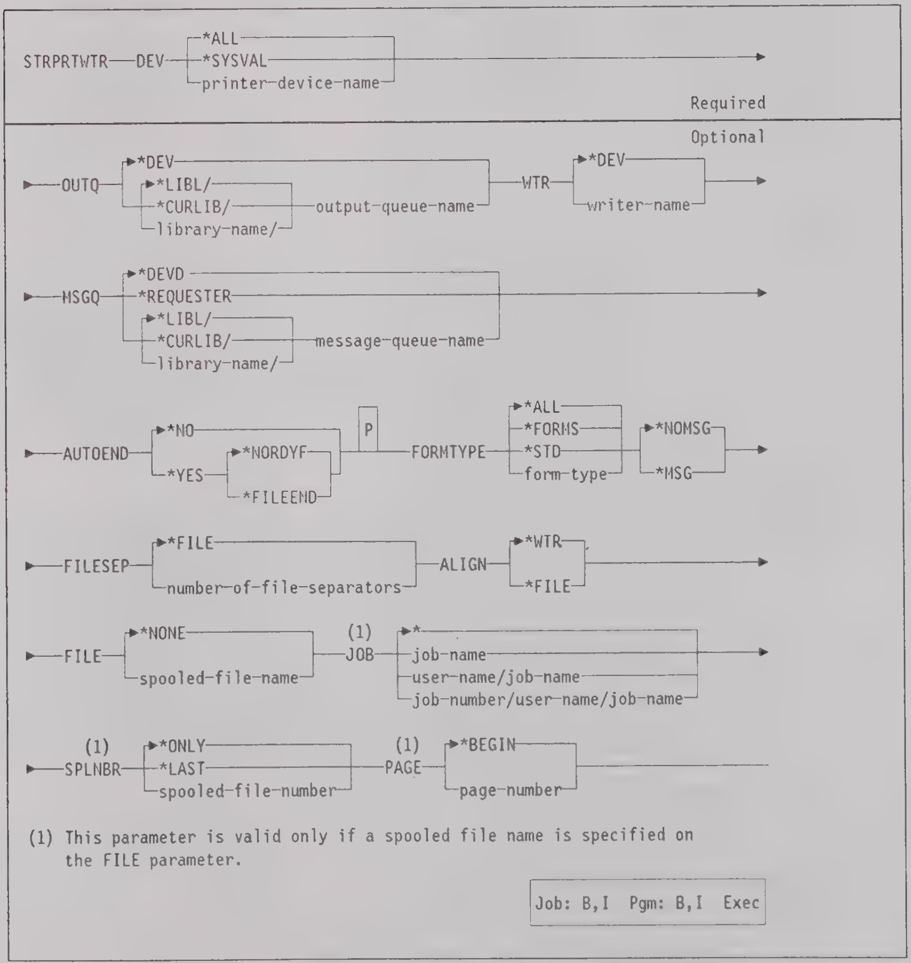
This command starts a batch job by accepting data from the device and printing the data in printer file QPEMPRTF. The output prints immediately; the maximum number of lines per page is 96. If a timeout of 10 minutes occurs, printer emulation forces out the emulation output.

STRPRTWTR (Start Printer Writer) Command

The Start Printer Writer (STRPRTWTR) command starts a spooling writer to the specified printer. The writer, which is a system job, takes spooled output files from an output queue and produces (writes) the output on the printer device. This command specifies the name of the printer, the names of the output and message queues used, and the name of the writer.

More than one writer can be active at the same time (as determined by the spooling subsystem description). Each writer must have a unique writer name, its own output queue, and its own device. A writer that has been started can be actively writing output or waiting for a file to be put on the output queue. Optionally, the writer can end automatically when it has processed all the files on the output queue. The writer can also be changed, held, or canceled if the Change Writer (CHGWTR), Hold Writer (HLDWTR) or End Writer (ENDWTR) command is used.

Because each writer runs independently of the job that started it, the user can continue doing other work on the system after he has started a writer. The writer is owned by the user who issued the STRPRTWTR command.



(1) This parameter is valid only if a spooled file name is specified on the FILE parameter.

Job: B,I Pgm: B,I Exec

DEV Parameter

Specifies the name of the printer device used to print the spooled output. The device must be available for allocation before the writer can be started.

***ALL:** Start a print writer to every printer configured on the system.

***SYSVAL:** Start a print writer to the system default printer.

printer-device-name: Specify the name that the printer device being started is identified by.

OUTQ Parameter

Specifies the qualified name of the output queue from which the writer processes spooled output files.

***DEV:** Use the default output queue associated with the printer specified on the DEV parameter.

The possible library values are:

***LIBL:** The library list is used to locate the output queue.

***CURLIB:** The current library for the job is used to locate the output queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the output queue is located.

output-queue-name: Specify the name of the output queue that the writer processes output files from.

WTR Parameter

Specifies the name of the spooling writer being started. Each writer name must be unique.

***DEV:** The name of the writer is the same as the name of the printer device specified in the DEV parameter.

writer-name: Specify the name that the writer being started is identified by.

MSGQ Parameter

Specifies the qualified name of the message queue that messages created by the writer are sent to.

***DEVD:** The messages are sent to the message queue specified in the device description of the device named in the DEV parameter.

***REQUESTER:** The messages are sent to the message queue of the user who started the writer. This value becomes *DEVD for batch jobs.

The possible library values are:

***LIBL:** The library list is used to locate the message queue.

***CURLIB:** The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message queue is located.

message-queue-name: Specify the qualified name of the message queue that messages created by the writer are sent to.

AUTOEND Parameter

Specifies whether the writer ends automatically.

***NO:** The writer does not end when the last available file has been removed from the output queue; it waits for another spooled file entry to be put on the queue.

***YES:** The writer automatically ends after it has reached the state specified by the second part of the parameter (*NORDYF or *FILEEND).

***NORDYF:** The writer automatically ends when there are no ready files (all the available files have been removed from the output queue).

***FILEEND:** The writer stops after it has finished processing one spooled output file.

FORMTYPE Parameter

Specifies the form type selection values for which spooled files are to be produced by the writer. A file's form type is specified in the device file that produced the output file. This parameter specifies that only the files with this form type are processed now. All other files are left on the output queue as available.

To change the type of form for which spooled files are being produced, issue the Change Writer (CHGWTR) command.

***ALL:** All form types are processed by the writer.

***FORMS:** All available files on the output queue with the same form type are processed as a group before the writer moves on to the next form type. The writer first chooses the first available file on the queue. After the first file is complete, all files with the same form type will be processed. The writer again chooses the first available file on the queue and repeats the process for that form type.

***STD:** The writer is to produce spooled files with a form type of *STD.

form-type: Specify the type of form that you want your spooled files produced for.

***NOMSG:** The writer does not send an informational message when it runs out of files with the specified form type.

***MSG:** The writer sends an informational message when it runs out of files with the specified form type. If the writer does not end (AUTOEND parameter on STRPRTWTR command), then, as additional spooled files become available for printing, they are processed. The informational message is sent again each time the writer must wait for additional spooled files.

FILESEP Parameter

Specifies how to control the number of file separator pages that are printed before each file.

***FILE:** The number of separators that was specified for each individual file is used.

number-of-file-separators: Specify the the number (range of 0 to 9) of separator pages to print. Whenever you respond to the change form type message indicating that a new form type has been put on the printer, the writer issues a message inquiring how many file separator pages should be printed with the new form type.

ALIGN Parameter

Specifies how to control the forms alignment.

WTR: The writer keeps track of the output that is printed and issues a forms alignment message whenever it determines that forms may need to be aligned.

FILE: The forms alignment message is issued for every file that has ALIGN(*YES) specified. This option should be taken whenever the automatic forms alignment control provided by the writer does not provide the desired results.

FILE Parameter

Specifies the name of the first (or only) spooled output file to process by the spooling writer and print on the printer. If several files are available on the output queue, the next file produced is the first one available with the highest priority.

NONE: No spooled file name is specified; the first spooled file that becomes available on the output queue is processed first.

spooled-file-name: Specify the name of the spooled output file that is the first (or only) output file to be written to the printer.

JOB Parameter

Specifies the name of the job that created the spooled file. This parameter is valid only if a spooled file name is specified in the FILE parameter.

***:** The job from which this STRPRTWTR command was entered is the job that created the spooled file.

job-name: Specify the name of the job that created the spooled file. If no job qualifier is given, all of the jobs currently in the system are searched for the simple job name. (For an expanded description of the JOB parameter and duplicate job names, see Appendix A, "Expanded Parameter Descriptions" in Volume 1.)

SPLNBR Parameter

Specifies the number of the spooled output file to be processed first. This parameter is valid only if a spooled file name is specified in the FILE parameter. (For an expanded description of the SPLNBR parameter, see Appendix A, "Expanded Parameter Descriptions" in Volume 1.)

ONLY: Only one spooled output file from the job on the specified output queue has the specified file name; no spooled file number is needed. If *ONLY is specified and more than one spooled file on the output queue has the specified name, an error message is displayed to the user.

LAST: The highest-numbered spooled output file for the job with the specified file name on the specified output queue is the file processed first.

spooled-file-number: Specify the number of the spooled file for the job on the specified output queue that is being processed first.

PAGE Parameter

Specifies the page number of the first page to print from the first file.

BEGIN: The first page in the file is the first page to print.

page-number: Specify the number of the first page to print.

Example

```
STRPRTWTR  DEV(QSYSPRT)  OUTQ(QPRINTS)  WTR(TOM)
```

This command starts a spooling writer named TOM. This writer takes the output from the output queue named QPRINTS and prints the output on the printer named QSYSPRT. Any writer messages are sent to the system operator's message queue, and the writer waits for more output when the queue is emptied.

STRQRY (Start Query) Command

The Start Query (STRQRY) command shows a user the main Query menu. You must have Query on your system to run this command.

```
                                                          Optional
STRQRY ──────────────────────────────────────────────────────────

                                                 ┌─────────────────────┐
                                                 │Job: I  Pgm: I  Exec │
                                                 └─────────────────────┘
```

There are no parameters for this command.

Example

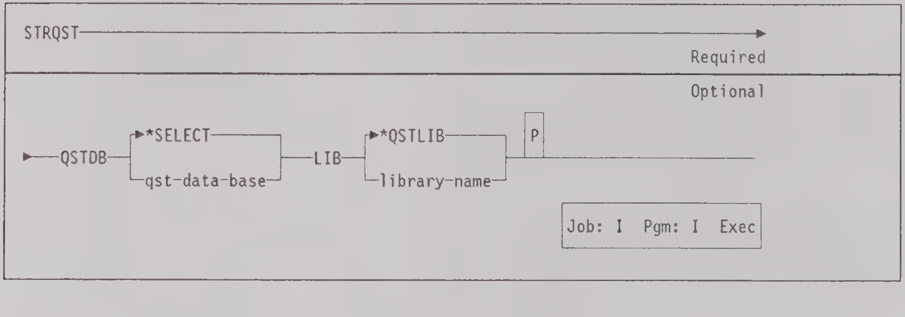
```
STRQRY
```

This command shows the main Query menu.

STRQST (Start Question and Answer) Command

The Start Question and Answer (STRQST) command shows a user the main Question & Answer menu.

Restriction: This command only runs interactively.



QSTDB Parameter

Specify the Question-and-Answer (Q & A) data base in which you want to work.

***SELECT:** You will be prompted to enter or select a Q & A data base. If only one Q & A data base exists on your system, it will be selected for you.

qst-data-base: Specify the Q & A data base that you want to use.

LIB Parameter

Specifies the name of the library that contains the Q & A data base.

***QSTLIB:** The library containing the specified Q & A data base is used.

library-name: Specify the library where the Q & A data base is located.

Example

```
STRQST
```

This command shows the Question and Answer (Q & A) main menu.

STRRJECSL (Start RJE Console) Command

The Start RJE Console (STRRJECSL) command shows the RJE console at the work station where the STRRJECSL command is sent. This command is sent interactively from an AS/400 work station.

RJE and host system messages appear on the console. You can enter host system commands on the command line to send to the host system if the RJE session is active.

The RJE session is not active when this command is sent.

Restriction: To use this command, you must have object operational authority to the session description and read authority to the library where the session description is stored.

```
                                                                Required
                       ┌►*LIBL/───────┐
STRRJECSL───SSND───┼─*CURLIB/──────┼──session-description-name ──────────────
                       └─library-name/─┘

                                                          ┌─────────────────┐
                                                          │Job: I  Pgm: I  Exec│
                                                          └─────────────────┘
```

SSND Parameter
Specifies the qualified name of the session description associated with the RJE message queue that interactively shows the host system messages. If the RJE session is active, you can enter host system commands.

The possible library values are:

***LIBL:** The library list is used to locate the session description.

***CURLIB:** The current library for the job is used to locate the session description. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the session description is located.

Example

```
STRRJECSL  SSND(USERLIB/RJE)
```

This command starts a RJE console. Messages from the RJE message queue specified in the session description named RJE in library USERLIB are displayed. If the RJE session is active, commands can be sent to the host system.

STRRJERDR (Start RJE Reader) Command

The Start RJE Reader (STRRJERDR) command starts an RJE reader to send data to the host system.

The RJE job queue in the session-description reader entry is released.

If an RJE job queue is not specified in the session-description reader entry, the reader is reserved for interactive jobs.

The command sends data to the host system to start the host reader. If the Start RJE Session (STRRJESSN) command is sent with RDRS(*NO) and SNDSTRCMD(*NO), a start command is not sent the first time you start the reader in the session.

Restriction: To use this command, you must have object operational authority to the session description.

```
                  ┌►*ALL───────┐
STRRJERDR──RDR──┤            ├──SSN remote-job-entry-session-name──────────
                  └─RD1 - RD15─┘

                                                  ┌─────────────────────────┐
                                                  │Job: B,I  Pgm: B,I  Exec│
                                                  └─────────────────────────┘
```

RDR Parameter
Specifies the RJE reader that is started.

*ALL: Specifies all RJE readers in the RJE session.

RD1 - RD15: Specify a value from 1-15 for a specified reader, for example, RD3.

Note: *AUTO is started (its job queue released, when specified) when a reader is started that uses *AUTO.

SSN Parameter
Specifies the active RJE session associated with the RJE reader that sends RJE reader jobs to the host system.

Example

```
STRRJERDR  RDR(RD1)  SSN(RJE)
```

This command starts RD1 (reader 1) in the active RJE session name RJE. Starting a reader allows it to use the Submit RJE Job (SBMRJEJOB) command to send files to the host system.

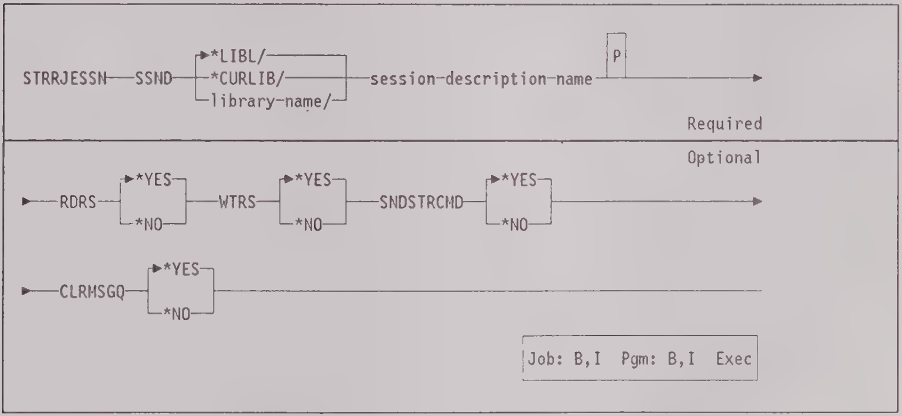
STRRJESSN (Start RJE Session) Command

The Start RJE Session (STRRJESSN) command starts a Remote Job Entry (RJE) session. Each session has a unique name.

Note: When RJE readers and/or writers are started, start commands are sent to the host system to start the associated host functions. If the host system reader or writer is already started, some host systems return error messages. RJE ignores these messages.

If SNDSTRCMD(*NO) is specified, the start commands for the specified readers and writer are not sent. If SNDSTRCMD(*YES) is specified, only the start commands for the readers and writers being started are sent.

Restriction: To use this command, you must have object operational authority to the session description, read authority to the library where the session description is stored, and control authority to the job.



SSND Parameter
Specifies the qualified name of the session description for the session that is started.

The possible library values are:

*LIBL: The library list is used to locate the session description.

***CURLIB:** The current library for the job is used to locate the session description. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the session description is located.

RDRS Parameter
Specifies if the RJE readers defined in the session description are started.

*YES: RJE readers are started. (That is, all reader job queues defined in the session description are released by RJE and a START command is sent to the host system.) Only readers with corresponding communications entries are started. Also, the corresponding communications devices must be varied on, or the session is not started.

***NO:** RJE readers are not started. The Start RJE Reader (STRRJERDR) command starts the readers after the session is active.

WTRS Parameter
Specifies if the RJE writers defined in the session description are started.

*YES: RJE writers are started. Corresponding communications devices must be varied on, or the session is not started.

***NO:** RJE writers are not started. If a request is received from the host system for an RJE writer that is not started, an informational message is sent to the RJE message queue. The data received is held until the RJE writer is started. The Start RJE Writer (STRRJEWTR) command starts the writers after the session is active.

SNDSTRCMD Parameter
Specifies if the start reader and writer commands are sent to the host when the readers and writers start for the first time after the RJE session is started.

*YES: The start commands are sent.

***NO:** The start commands are not sent. The start commands *are* sent to the host system if the readers and writers are canceled and restarted in the same session. SNDSTRCMD(*NO) is used when the RJE session logon automatically sends the start reader and writer commands.

CLRMSGQ Parameter

Specifies if the messages from previous sessions are printed and cleared from the session description message queue.

*YES: Messages are printed and cleared.

***NO:** Messages are not printed and cleared.

Example

```
STRRJESSN  SSND(USERLIB/RJE)  RDRS(*YES)  WTRS(*YES)
```

This command starts a RJE session. The session description named RJE in library USERLIB describes the RJE session environment. In addition to starting the required RJE jobs (session control, console input, and console output), readers and writers (printers and punches) defined in the session description are started. When the RJE session has started, read files are sent to the host system and host system files are received by the writers. Host system commands are sent by starting the RJE console.

STRRJEWTR (Start RJE Writer) Command

The Start RJE Writer (STRRJEWTR) command starts all Remote Job Entry (RJE) writers or a specific RJE writer to receive writer output from the host system.

If the host system writer is already started, some host systems return error messages. RJE ignores these messages. If the Start RJE Session (STRRJESSN) command is sent with WTRS(*NO) and SNDSTRCMD(*NO), a start command is not sent to the host system the first time you start the writer.

Restrictions: To use this command, you must have object operational authority to the session description.

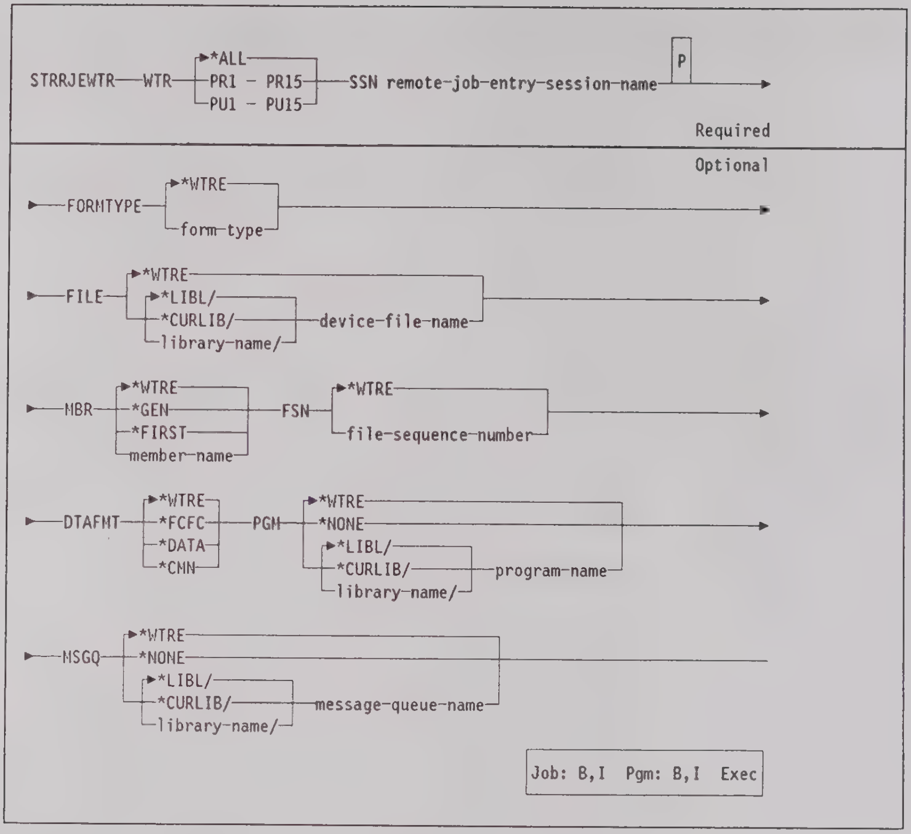
The RJE user profile (QRJE) must have object operational authority to any program specified in this command and read authority to the library where the specified program is stored.

If a data base file name is specified in this command or in the session-description RJE writer entry (that is, if the WTR parameter of this command defaults to the value specified in the RJE writer entry), the RJE user profile must have add authority to that data base file.

If MBR(*GEN) is specified or is in the session-description RJE writer entry, the RJE user profile must have object management and object operational authority to the data base file as well as read authority to the library where the data base file is stored.

If a device file name is specified in this command or in the session-description RJE writer entry (that is, if the FILE parameter of this command defaults to the value specified in the RJE writer entry), the RJE user profile must have object operational authority to that device file and read authority to the library where the device file is stored.

If a message queue is specified in the session-description RJE writer entry (if the MSGQ parameter of this command defaults to the value specified in the RJE writer entry), the RJE user profile must have objects operational authority to that message queue and read authority to the library where the message queue is stored.



WTR Parameter
Specifies the RJE writer that is started.

ALL: Specifies all RJE writers in the RJE session.

PR1 - PR15: Specify a value from 1 through 15 for the printer that is started, for example, PR3.

PU1 - PU15: Specify a value from 1 through 15 for the punch device that is started, for example, PU3.

SSN Parameter
Specifies the name of the active RJE session that contains the RJE writer that is started.

FORMTYPE Parameter
Specifies the initial form type.

***WTRE:** The form type in the session-description writer entry.

form-type: Specify the initial form type. Valid entries are 1 to 8 alphameric characters in length. The value corresponds to a valid entry in the forms control table (FCT) you specified in the Create Session Description (CRTSSND) command.

FILE Parameter
Specifies the qualified name of the RJE writer device file (printer only) or the AS/400 data base file that receives output data from the host system.

***WTRE:** Specify the device file in the session-description RJE writer entry.

The possible library values are:

***LIBL:** The library list is used to locate the file.

***CURLIB:** The current library for the job is used to locate the file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the file is located.

device-file-name: Specify the name of the program-described device file.

MBR Parameter
Specifies the data base file member the output is directed to.

***WTRE:** Creates data base file members according to the method specified in the session-description writer entry.

***GEN:** Creates a member name. For example,

```
Afffffccc or Bffffffccc
```

Where:

A	Indicates that the file member contains print data.
B	Indicates that the file member contains punch data.

ffffff	Stands for 1 to 6 characters of the forms name specified in the FCT or received from the host system. Only characters that are valid in an AS/400 name are valid in the form type that creates data base file member names.
ccc	Stands for a three-digit sequence value controlled by the RJE session to maintain member uniqueness.

If a member name already exists in the data base file, the three-digit sequence value increases by one. RJE increases the sequence value until a unique name is reached and a member is created, or until RJE tries all 1000 possibilities without creating a member.

If a member is removed by the Remove Member (RMVM) command, the member of the same sequence number is not created again unless you reset sequence numbering with the Start RJE Writer (STRRJEWTR) command.

If no member is created, the RJE operator receives a message indicating the failure and a request to either try again or cancel this file.

You cannot create members in a Data Distributed Management (DDM) file.

***FIRST:** Specifies the first member of the data base file.

member-name: Specify the name of the data base file member. If the member does not exist, an inquiry message is sent to the RJE message queue.

FSN Parameter

Specifies the initial three-digit file sequence number that creates data base file member names. This parameter applies only if you specified *GEN for the MBR parameter.

***WTRE:** The file sequence number in the session-description writer entry.

file-sequence-number: Specify the initial three-digit file sequence number. Leading zeros are not required for sequence numbers less than 100.

DTAFMT Parameter

Specifies the format of the output data. This parameter applies only to print data.

***WTRE:** The output format from the session-description writer entry.

***FCFC:** The output data is in the first-character forms control (FCFC) data format (in 133- or 199-character records). The first character of each record is the printer American National Standards Institute (ANSI) forms control information. If DEVTYPE(*PUN) is specified, *FCFC is not valid.

To write the data to a data base file in the FCFC data format, send the Copy File (CPYF) command and specify a printer file for the TOFILE parameter. This procedure lets you print the file later.

***DATA:** The output data is in the normal data format (in 132- or 198-character records). No FCFC characters are embedded in the data. Specify *DATA to send data to a data base file to be processed by a program. If data is directed to a printer file, a single-space ANSI control character becomes the first character in each record.

***CMN:** The output data is in the communications data format (in 256-character records). The communications format is either compressed or truncated. *CMN decreases communications time. However, before you use the data, you must use the Convert RJE Data (CVTRJEDTA) command to change the data to *FCFC or *DATA. If you specify *CMN, the output file must be a data base file with a length of 256.

PGM Parameter

Specifies the qualified name of a user-supplied program associated with the RJE writer.

***WTRE:** Specify the associated session-description writer entry.

***NONE:** A user-supplied program is not associated with the RJE writer.

The possible library values are:

***LIBL:** The library list is used to locate the user-supplied program.

***CURLIB:** The current library for the job is used to locate the user-supplied program. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the user-defined program is located.

program-name: Specify the name of the user-supplied program.

MSGQ Parameter

Specifies the qualified name of the user message queue where messages for the RJE writer are recorded.

Note: Messages for RJE writers are recorded in the RJE message queue associated with the RJE session. The RJE message queue name depends on the name specified in the MSGQ parameter in the Create Session Description (CRTSSND) or Change Session Description (CHGSSND) command. If inquiry messages are sent by RJE, they are sent to the user message queue (if specified) where they must receive a response.

***WTRE:** Specify the user message queue in the session-description writer entry.

***NONE:** A user message queue is not used.

The possible library values are:

***LIBL:** The library list is used to locate the message queue.

***CURLIB:** The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message queue is located.

message-queue-name: Specify the name of the user message queue.

Example

```
STRRJEWTR  WTR(*ALL)  SSN(RJE)  FORMTYPE(COMMON)  +
  MSGQ(USERLIB/BROWN)
```

This command starts all RJE writers not already started in an active RJE session named RJE. Each writer started with the formtype in the session description writer entry is overridden and changes to formtype COMMON. This forms control entry is used for each writer that the host system does not send a forms mount message to and that refers to a valid forms control entry (assuming the session description specified a forms control table name).

STRSAM (Start Sampled Address Monitor) Command

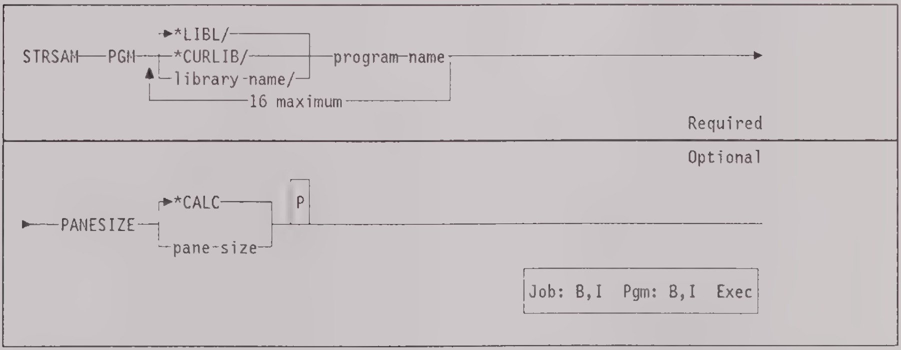
The Start Sampled Address Monitor (STRSAM) command prepares to periodically sample the Machine Interface (MI) instruction address while running selected programs, to produce an approximation of the time spent running each one of the program statements. The sampling environment it defines remains in effect until the End Sampled Address Monitor (ENDSAM) command is issued, even across IPLs. A new environment may not be defined while an old one is still in effect.

Once the environment is defined, the Start Sampled Address Monitor Data Collection (STRSAMCOL) command is used to start the data collection: the system periodically samples the currently running program; if it is one of the selected programs (regardless of which job is running it), the system increments a counter corresponding to the current MI instruction address within that program. After the selected programs are run, the End Sampled Address Monitor Data Collection (ENDSAMCOL) command is used to turn sampling off and optionally save in a data base file any sample data collected. Sampling may be turned on and off any number of times. The results may or may not be cumulative, depending on the parameter values used.

The Print Sampled Address Monitor Data (PRTSAMDTA) command produces reports used to analyze the sample data stored in the data base file. This report is used to determine the relative percentage of processing time spent in different parts of the selected programs.

All sampled address monitor internal data (environment definition and counters) is system-wide, not private; therefore, this feature may not be used at the same time by more than one job.

The STRSAM command is part of the IBM AS/400 Performance Tools product, Program 5728-PT1. More information on the performance tools is in the *Performance Tools Guide*.



PGM Parameter
Specifies a list of up to 16 programs to select for sampling.

program-name: Specify the name of a program to select.

The possible library values are:

***LIBL:** The library list is used to locate the program.

***CURLIB:** The current library for the job is used to locate the program. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the name of the library where the program is located.

PANESIZE Parameter
Specifies the pane size, which is the number of consecutive program instruction addresses assigned to each counter. The smaller the pane size, the more accurately the reports can pinpoint program activity.

***CALC:** For each selected program, the optimal pane size for a high-level language program is computed based on the size of that program. The result is such that there are more than 2048 panes (counters) but not more than 4096 panes for each program, except a computed pane size less than 4 is increased to 4.

pane-size: Specify one pane size to use for all selected programs. The size must be power of 2 from 2 through 4096: 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, or 4096. A pane size may need to be selected for large programs, where the computed pane size may be too large to provide enough detail.

Examples

```
STRSAM   PGM(MYPROG)   PANESIZE(4)
```

This command defines a sampling environment for the program *LIBL/MYPROG. Every sample counter maps to four instruction addresses, which is generally adequate for analyzing a high-level language program.

```
STRSAM   PGM(PROG1 PROG2)
```

This command defines a sampling environment for two programs: *LIBL/PROG1 and *LIBL PROG2. An optimal pane size will be computed for each program.

```
STRSAM   PGM(PROG1 PROG2)   PANESIZE(2)
```

This command defines a sampling environment as in the previous example, except that a pane size of 2 is used for each program.

STRSAMCOL (Start Sampled Address Monitor Data Collection) Command

The Start Sampled Address Monitor Data Collection (STRSAMCOL) command defines a sample interval and begins collecting sample data as previously defined by the Start Sampled Address Monitor (STRSAM) command. Sampling continues, even across IPLs, until stopped by the End Sampled Address Monitor Collection (ENDSAMCOL) command.

When sampling is active, system performance degrades exponentially relative to the inverse of the sample interval. This degradation is approximately 40% at the shortest interval of one millisecond. It decreases as the sample interval increases: 10% for a 5 millisecond sample interval, 5% for 10 millisecond, etc.

The STRSAMCOL command is part of the IBM AS/400 Performance Tools product, Program 5728-PT1. More information on the performance tools is in the *Performance Tools Guide.*

```
STRSAMCOL──────────────────────────────────────────────────────────────►
                                                               Required
─────────────────────────────────────────────────────────────────────────
                                                               Optional
               ┌►1 ──────────────────────┐        ┌►*YES┐
►──INTERVAL────┤                         ├──CLEAR─┤     ├──────────────
               └─number-of-milliseconds──┘        └─*NO─┘
                                                  ┌──────────────────────┐
                                                  │Job: B,I  Pgm: B,I Exec│
                                                  └──────────────────────┘
```

INTERVAL Parameter

Specifies the instruction address sample interval in number of milliseconds between samples.

1: A 1 millisecond sample interval will be used. This is the shortest interval allowed. It provides the most accurate results, but also severely impacts system performance.

number-of-milliseconds: Specify the number of milliseconds between samples, from 1 through 63. Shorter intervals provide more accurate results. Longer intervals impact the system performance less.

CLEAR Parameter

Specifies the disposition of any previously accumulated internal instruction address counters.

***YES:** Any previously collected data is cleared.

***NO:** Any previously collected data is kept.

If the CLEAR parameter is specified with a value of *NO, the INTERVAL parameter value used should be the same as that used with the last STRSAMCOL command (unless no samples were taken at the old value); if not, the report created by PRTSAMDTA will be of little or no value.

Examples

```
STRSAMCOL   INTERVAL(5)
```

This command begins sampling every 5 milliseconds. Whenever a sample is taken and one of the programs defined by STRSAM is running, the counter for the correct pane is incremented.

```
STRSAMCOL   CLEAR(*NO)
```

This command continues sampling, after an intervening ENDSAMCOL, without clearing the previously collected data.

STRSBS (Start Subsystem) Command

The Start Subsystem (STRSBS) command starts a subsystem based on the subsystem description specified in the command. Once started, the subsystem is known by the unqualified name of the subsystem description (the subsystem description name without the library name). When the subsystem is started, the system allocates the necessary and available resources (storage, work stations, and job queue) that are specified in the subsystem description.

Storage is allocated to the subsystem according to the storage pool definitions specified in the subsystem description, starting with the lower numbered storage pool definitions. If all the pool definitions cannot be allocated, because the maximum number of storage pools on the system is reached or because insufficient storage is available, messages indicating which pools could not be allocated are sent to the system operator. If storage becomes available later, or if the number of active storage pools is reduced, the available resources are automatically allocated to the subsystem to satisfy its unfulfilled requirements. Any jobs that would normally run in a storage pool that is not allocated are run in the shared storage pool (*BASE).

Allocating Work Stations: Work stations are allocated to the subsystem according to the work station entries in the subsystem description. Each work station whose name (or type, if not specified by name) is contained in one of the subsystem description's work station entries, and whose entry specifies AT(*SIGNON), is allocated to this subsystem unless it is currently signed on to another subsystem. The sign-on prompt is displayed on each work station that is allocated. Work stations that are already signed on in another subsystem remain allocated to that subsystem until the subsystem that allowed the sign-on is ended, or until you transfer your job to this subsystem. (Messages indicating the names of the work stations that could not be allocated are sent to the system operator.)

If multiple subsystems specify the same work station in their work station entries, each subsystem, as it is started, attempts to allocate that work station. Each successive subsystem allocates that work station unless a user signs on while the work station is allocated to one of the previously started subsystems. When a signed-on work station is signed off, it still remains allocated to the same subsystem until another subsystem is started that specifies that work station. If, however, a work station is varied offline and several active subsystems specify that work station, the subsystem to which the work station is allocated when it is varied online is unpredictable.

Allocating Job Queues: If a job queue is specified in the work entries of the subsystem description, the job queue is allocated to the subsystem. If the job queue does not exist or if it is already allocated to an active subsystem, no job queue is allocated to the subsystem and a message is sent to the system operator. If the job queue later becomes available, it is *not* automatically allocated to the subsystem. To allocate the job queue to the subsystem, the subsystem must end, then start again.

Restriction: To start a subsystem, you must have object operational authority to use the subsystem description.

```
                                                                    Required

                  ┌►*LIBL/──────┐                          │P│
STRSBS ── SBSD ───┼─*CURLIB/────┴──subsystem-description-name─┴─────────

                  └─library-name/─
                                                          ┌──────────────────────┐
                                                          Job: B,I  Pgm: B,I  Exec
                                                          └──────────────────────┘
```

SBSD Parameter
Specifies the qualified name of the subsystem description that defines the operational environment (subsystem) being started.

The possible library values are:

***LIBL:** The library list is used to locate the subsystem description name.

***CURLIB:** The current library for the job is used to locate the subsystem description name. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the subsystem description name is located.

The name of the subsystem description cannot be the same as the name of a subsystem that is currently active, even though the subsystem descriptions are in different libraries.

Examples

```
STRSBS  SBSD(QBATCH)
```

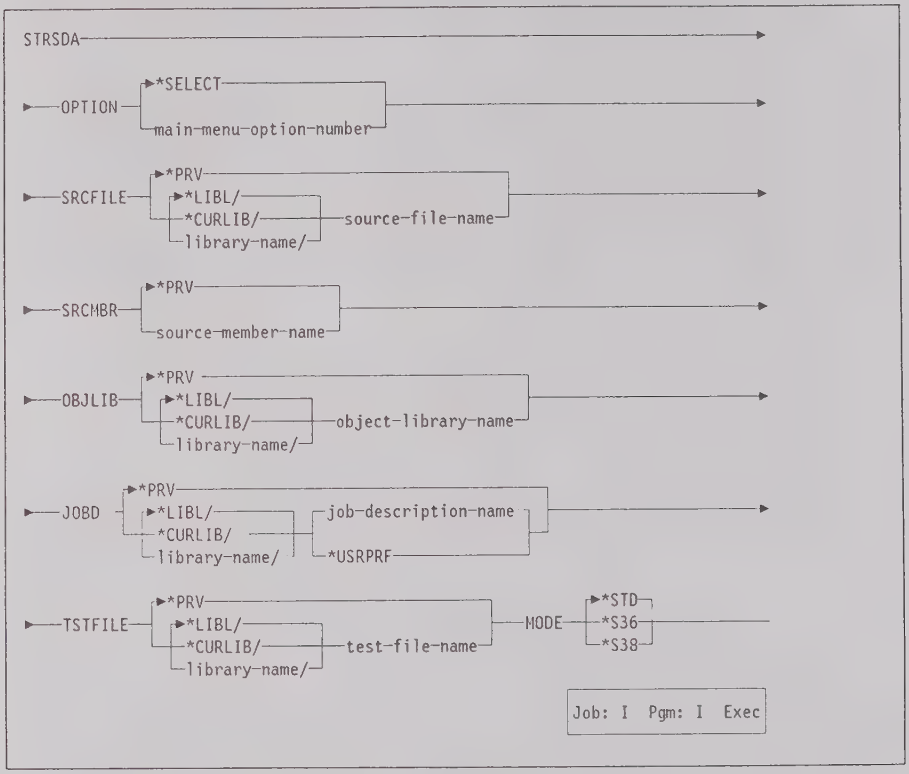
This command starts the batch subsystem named QBATCH.

```
STRSBS  SBSD(QGPL/TELLER)
```

This command starts the subsystem that is associated with the TELLER subsystem description in library QGPL. The subsystem name is TELLER.

STRSDA (Start SDA) Command

The Start SDA (STRSDA) command is the primary command for the IBM AS/400 Design Aid (SDA) utility. This command can be called in the System/36 and System/38 environments, as well as the AS/400 operating system.



OPTION Parameter
Specifies which option to use as a value for the SDA main menu. This parameter is ignored if MODE (*S36) is specified.

SELECT: Displays the SDA main menu.

main-menu-option-number: Specify a number between 1 and 3 that corresponds to an option on the SDA main menu. If you select this parameter value, the SDA main menu will not appear.

SRCFILE Parameter
Specifies the qualified name of the source file that contains the source member updated or to which a new source member is added. Only the library qualifier is used if MODE (*S36) is specified.

PRV: Specifies that SDA uses the name of the source file and library used in your last SDA session for the AS/400 system (only when MODE is (*STD)).

The possible library values are:

LIBL: The library list is used to locate the name of the source file.

***CURLIB:** The current library for the job is used to locate the source file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the source file is located.

source-file-name: Specify the qualified name of an existing source file used by SDA.

SRCMBR Parameter
Specifies the name of a new or existing source file member that contains or will contain source for the displays or menus updated or created by SDA.

PRV: Specifies that SDA is to use the name of the source member used in your last SDA session for the AS/400 system (only when MODE is (*STD)).

source-member-name: Specify the name of the source member created or updated.

OBJLIB Parameter
Specifies the qualified name of the library into which the program or display file that SDA creates is stored.

***PRV:** Specifies that SDA uses the name of the object library used in your last SDA session for the AS/400 system (only when MODE is (*STD)).

The possible library values are:

***LIBL:** The library list is used to locate the name of the library the program or display file is stored.

***CURLIB:** The current library for the job is used to locate the SDA program. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the SDA program is located.

object-library-name: Specify the name of the library in which objects created by SDA are stored.

JOBD Parameter

Specifies the qualified name of the job description used with batch jobs being submitted by SDA. This parameter is not used if MODE (*S36) is specified.

***PRV:** Specifies that SDA uses the name of the job description and library used in your last SDA session for the AS/400 system (only when MODE is (*STD)).

The possible library values are:

***LIBL:** The library list is used to locate the job description.

***CURLIB:** The current library for the job is used to locate the job description. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the job description is located.

job-description-name: Specify the qualified name of the job description used with submitted jobs.

*USRPRF: Specifies that SDA uses the name of the job description defined in the user profile.

TSTFILE Parameter

Specifies the qualified name of the display file used for testing. This parameter is ignored if MODE (*S36) is specified.

***PRV:** Specifies that SDA uses the name of the display file and library used in your last SDA session for the AS/400 system (only when MODE is (*STD)).

The possible library values are:

***LIBL:** The library list is used to locate the name of the display file.

***CURLIB:** The current library for the job is used to locate the name of the display file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the name of the display file is located.

test-file-name: Specify the qualified name of the display file used for testing.

MODE Parameter

Specifies if the System/36 or System/38 environments or the AS/400 operating system is used by SDA.

***STD** Specifies AS/400 SDA and displays the AS/400 SDA main menu. The main menu will not appear if the OPTION parameter is specified.

***S36:** Specifies the System/36 environment of SDA and displays the System/36 SDA main menu.

***S38:** Specifies the System/38 environment of SDA and displays the System/38 SDA main menu. The main menu will not appear if the OPTION parameter is specified.

Example

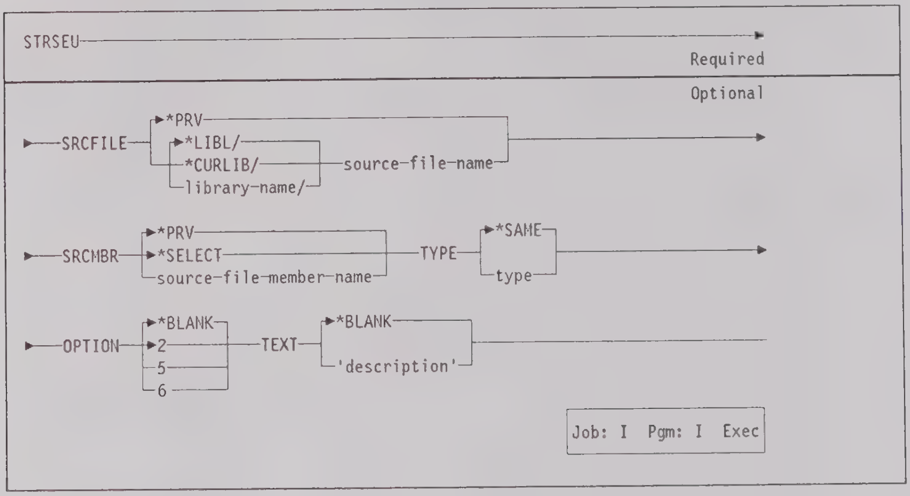
```
STRSDA  SRCFILE(TESTLIB/TESTFILE)  SRCMBR(TESTMBR)  +
  JOBD(*CURLIB/TESTJOB)
```

This command displays the SDA main menu. The source member to be created or updated is TESTMBR from the source file TESTFILE in the library TESTLIB. The name of the job description used with SDA batch job is TESTJOB in library *CURLIB. The defaults for all other parameters are assumed.

STRSEU (Start SEU) Command

The Start SEU (STRSEU) command allows the user to create, change, display, or print a source member.

The STRSEU command can create a source member but it cannot create a source file which contains source members. Use the Create Source Physical File (CRTSRCPF) command to create a source file, and use the Create Library (CRTLIB) command to create a library source file.



SRCFILE Parameter

Specifies the qualified names of the source library and source physical file that contains the member edited or created.

*PRV: Specifies the qualified source physical file and library name you used last.

The possible library values are:

*LIBL: The library list is used to locate the source file.

*CURLIB: The current library for the job is used to locate the source file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the source file is located.

source-file-name: Specify the name of an existing source file.

SRCMBR Parameter

Specify the names of the source physical file member edited or created. The default value for this parameter depends on if you specify the SRCFILE parameter.

*PRV: This is the default value if you do not specify the SRCFILE parameter. *PRV selects the name of the previous source member used.

*SELECT: This is the default value if you specify the SRCFILE parameter. Select a member to edit, display, print, or remove.

source-file-member-name: Specify the name of the member you are creating or editing.

TYPE Parameter

Specify the type of source that is edited or created.

*SAME: Specifies that the same type is used when the member was edited.

type: Specify the type of source used. You can specify any combination up to 10 characters or you can specify one of the following types supported by SEU:

- AS/400 types: BAS, BASP, CBL, CL, CLP, CMD, DSPF, ICFF, LF, MNU, PF, PLI, PRTF, RPG, RPT, AND TXT.

OPTION Parameter

Specify the function to perform on the selected member. The default value depends on if you specify the member name.

*BLANK: This is the default value if you do not specify a member name. *BLANK specifies no action.

The listed options allow you to select an operation as follows:

OPTION	OPERATION SELECTED
2	Edit a member (Option 2 is the default value)
5	Display a member

6	Print a member

TEXT Parameter

Specify a character string that describes the member and is stored in the text field for the member.

***BLANK:** For a new member, the default value directs SEU to specify blanks in the text field of the member. For an existing member, this default value does not change the text field of the member.

'description': Specify text no longer than 50 characters, enclosed in apostrophes.

Example

```
STRSEU  SRCFILE(QGPL/FILE1)  SRCMBR(TELEDIR)  (TYPE(CLD)  +
  TEXT('CL program to create office telephone directory')
```

This command uses a source file, FILE1, located in QGPL and contains a source member named TELEDIR. TELEDIR is a CL program, specified by TYPE(CLP), and is to be edited since the default value for the OPTION parameter is used (2 - edit a member). The TEXT parameter supplies a description of the member to be edited.

STRSPTN (Start Support Network) Command

The Start Support Network (STRSPTN) command opens a communications path through the IBM Information Network to the specified destination application.

This command is provided for customers who want to write their own programs to interface with one of the IBM support systems.

```
                                                                         Required
                                                                        ┌─┐
                                                                        │P│
STRSPTN──ACCOUNT account-number──SPTNUSRID support-network-user-ID──────┴─┴──────►

►──SPTNPWD support-network-password──FEADEV front-end-application-device──────►

►──DESTAPP destination-application─────────────────────────────────────────────
                                                               ┌──────────────┐
                                                               │Pgm: B,I      │
                                                               └──────────────┘
```

ACCOUNT Parameter
The support network organization account ID for the user. The organization account ID must be registered with the support network.

SPTNUSRID Parameter
The network user ID in the account specified in the ACCOUNT parameter. The user ID must be registered with the support network and must be valid for the specified account.

SPTNPWD Parameter
The network password for the specified user. The password must be registered with the support network and must be valid for the specified user.

Note: A support network password may expire. The user must then change his password interactively using the WRKPRDINF command.

FEADEV Parameter
The device description to be used as the Front End Application (FEA).

DESTAPP Parameter
Specifies the destination application (DESTA) that it is connected to.

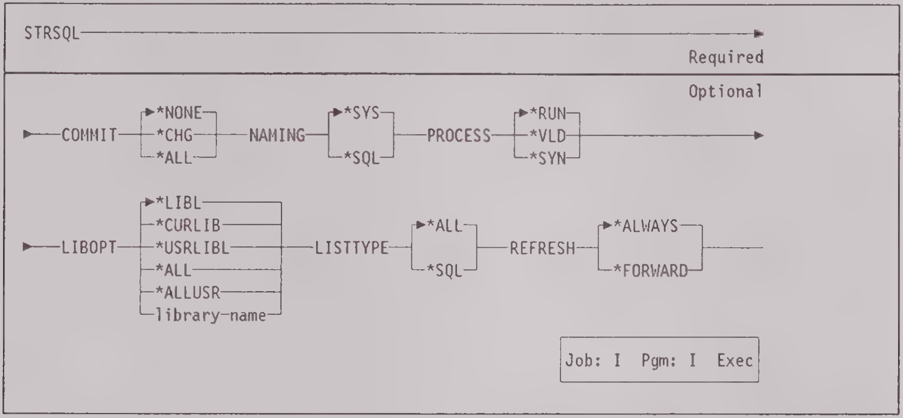
Example

```
STRSPTN  ACCOUNT(11420880)  SPTNUSRID(ACME)  SPTNPWD(11111)  +
         FEA(QTIFEA)  DESTA(AAAAAA)
```

This command establishes a communication path thru the IBM Information Network, for userid ACME at account 11420880. The path will allow access to the application AAAAAA.

STRSQL (Start Structured Query Language) Command

The Start Structured Query Language (STRSQL) command starts the interactive Structured Query Language (SQL) program. The program starts the statement entry of the interactive SQL program which immediately shows the Enter SQL Statements display. This display allows you to build, edit, enter, and run a SQL statement in an interactive environment. Any messages during the running of the program are shown on this display.



COMMIT Parameter

Specifies whether SQL statements are run under commitment control.

*NONE: Specifies that commitment control is not used. COMMIT and ROLLBACK statements are not allowed. If SQL DDL statements will be run, *NONE must be specified.

*CHG: Specifies that only those rows updated, deleted, or inserted are locked until the transaction is committed or rolled back.

*ALL: Specifies that every row selected, updated, deleted, and inserted are locked until the transaction is committed or rolled back.

NAMING Parameter

Specifies the naming convention used for naming objects in SQL statements.

*SYS: Indicates that the system naming convention is used (library-name/file-name).

*SQL: Indicates that the SQL naming convention is used (data-base-name.table-name).

PROCESS Parameter

Specifies what values are used to process the statements.

*RUN: Specifies that the statements are syntax checked, data checked, and then run.

*VLD: Specifies that the statements are syntax checked and data checked, but not run.

*SYN: Specifies that the statements are syntax checked only.

LIBOPT Parameter

Specifies which data bases and libraries are used as a basis for building a data base list when F16 is pressed.

*LIBL: All the libraries in the user and system portions of the job's library list are shown.

*CURLIB: The current library is shown.

*USRLIBL: Only the libraries in the user portion of the job's library list are shown.

*ALL: All of the libraries in the system, including QSYS, are shown.

*ALLUSR: All the nonsystem libraries, which include all user-defined libraries and the QGPL library, that are not in the job's library list are shown. Libraries whose names start with the letter Q, other that QGPL, are not included.

library-name: Specifies the name of the library.

LISTTYPE Parameter

Specifies what types of objects are displayed with list support (F16, F17, and F18).

*ALL: Specifies that all objects are displayed.

*SQL: Specifies that only SQL created objects are displayed.

REFRESH Parameter

Specifies when the display select output data is refreshed.

*ALWAYS: Specifies that the data is always refreshed.

***FORWARD:** Specifies that the data is refreshed only as the user scrolls forward to end of data for the first time. After that it is not refreshed.

Example

```
STRSQL COMMIT(*CHG) NAMING(*SQL)
```

This command starts an interactive SQL session that locks only those rows that are updated, deleted, and inserted until the transaction is committed or rolled back. The SQL naming convention is used for this session.

STRSRVJOB (Start Service Job) Command

The Start Service Job (STRSRVJOB) command starts the remote service operation for a specified job (other than the job issuing the command) so that other service commands can be entered to service the specified job. Any dump and trace commands can be run in that job until service operation ends. Service operation continues until the End Service Job (ENDSRVJOB) command is run.

The following commands will function in the serviced job while remote service is being done: Dump Job (DMPJOB), Dump Job Internal (DMPJOBINT), Dump Object (DMPOBJ), Dump System Object (DMPSYSOBJ), Trace Internal (TRCINT), and Trace Job (TRCJOB).

Restriction: To service a job, you must have operational authority for the user profile that the job is running under.

```
                                                          Required
                                                    ┌─┐
            ┌►JOB──────────────────────────────────┐│P│
STRSRVJOB───┼─job-name─────────────────────────────┼┴─┴──────────────────
            ├─user-name/job-name───────────────────┤
            └─job-number/user-name/job-name────────┘

                                                   ┌──────────────────────┐
                                                   │Job: B,I  Pgm: B,I  Exec│
                                                   └──────────────────────┘
```

JOB Parameter
Specifies the name of the job being serviced. If no job number is given, all of the jobs currently in the system are searched for the simple job name. If duplicates of the specified name are found, messages are sent to the user, and user name and job number must be specified. The job name entered cannot be the name of the job issuing the command. (For an expanded description of the JOB parameter, refer to Appendix A, "Expanded Parameter Descriptions.")

Example

```
STRSRVJOB   JOB(ABCD)
```

This command starts the remote service operation so that any trace or dump commands entered in this job are applied to the job named ABCD.

STRSST (Start System Service Tools) Command

The Start System Service Tools (STRSST) command shows the System Service Tools (SST) menu.

You can:

- Start a service function
- Work with disk unit configuration and data

Restriction: To start system service tools you must have special service authority.

```
                                                     Optional
STRSST--------------------------------------------------------

                                             Job: I  Pgm: I  Exec
```

There are no parameters for this command.

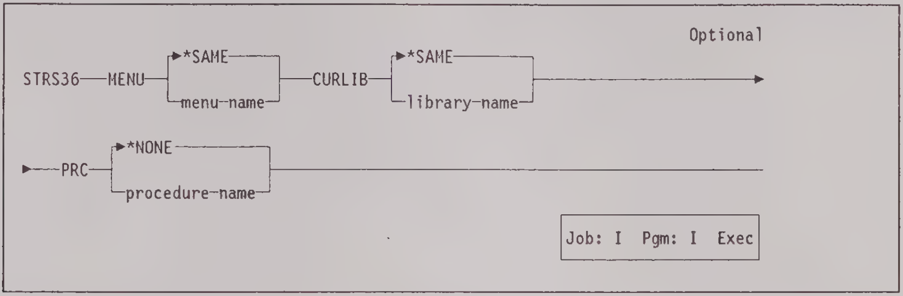
Example

```
STRSST
```

This command shows the Start System Service Tools menu.

STRS36 (Start System/36) Command

The Start System/36 (STRS36) command starts a System/36 environment session.



MENU Parameter
Specifies the first menu that is shown when the System/36 environment is started.

SAME: The MENU parameter is not changed, the menu that is currently being displayed will be redisplayed.

menu-name: Specify the menu name that is the first shown when the System/36 environment is started.

CURLIB Parameter
Specifies the current library to use in the System/36 environment.

SAME: The current library is not changed. If the current library is *CRTDFT and *SAME is specified, the current library is set to #LIBRARY.

library-name: Specify the name of the library you want to use for the current library in the System/36 environment.

PRC Parameter
Specifies the procedure to run when the System/36 environment is started.

NONE: No procedure runs when the System/36 environment is started.

procedure-name: Specify the name of the procedure to run when the System/36 environment is started.

Example

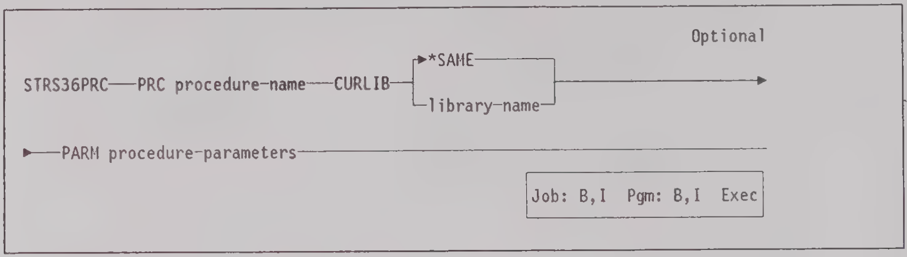
```
STRS36  MENU(user) CURLIB(MYLIB) PRC(INITPROC)
```

This command starts a System/36 environment session. This command:

- Displays user menu
- Changes the current library to MYLIB
- Runs procedure INITPROC

STRS36PRC (Start System/36 Procedure) Command

The Start System/36 Procedure (STRS36PRC) command starts a System/36 procedure.



PRC Parameter

Specifies the name of the System/36 procedure to run.

CURLIB Parameter

Specifies the current library to use to run the System/36 procedure.

<u>*SAME:</u> The current library is not changed. If the current library is *CRTDFT and *SAME is specified, the current library is set to #LIBRARY.

library-name: Specify the name of the library you want to use for the current library while running the System/36 procedure.

PARM Parameter

Specifies procedure parameters for the procedure. Procedure parameters allow information to be passed to the procedure. If no parameters are specified, no parameters are passed to the procedure.

Example

```
STRS36PRC  PRC(PROC1) CURLIB(MYLIB)
```

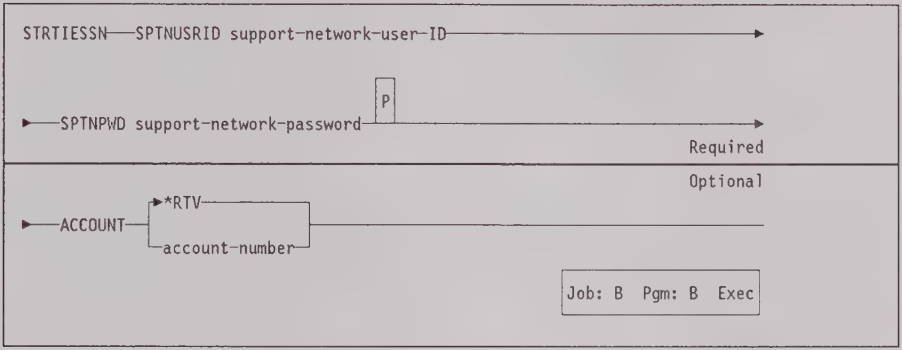
This command changes the current library to MYLIB and runs procedure PROC1.

```
STRS36PRC  PRC(CATALOG)  PARM('ALL,F1')
```

This command lists all files used by the System/36 Environment.

STRTIESSN (Start Technical Information Exchange Session) Command

The Start Technical Information Exchange Session (STRTIESSN) command establishes the communications link for a TIE batch session. This command must precede other TIE batch commands.



SPTNUSRID Parameter
Specifies the user ID used to signon to the IBM Information Network.

SPTNPWD Parameter
Specifies the password used to signon to the IBM Information Network.

ACCOUNT Parameter
Specifies the network account number used to sign-on to the IBM Information Network. If the account number is not specified, the account from the contact data base is used.

***RTV:** The account number from the contact data base is used.

account-number: Specify account number that is to be used.

Example

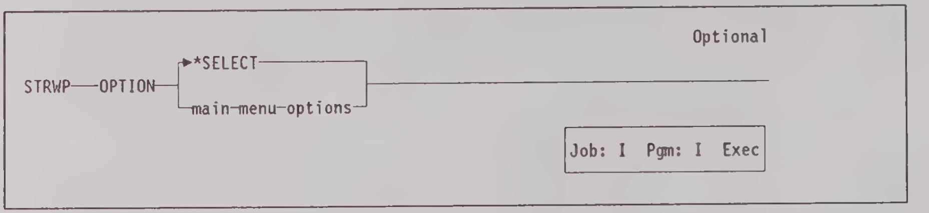
```
STRTIESSN  SPTNUSRID(ACME)   SPTNPWD(11111)  ACCOUNT(11420880)
```

This command displays the TIE main menu for account number 11420880.

STRWP (Start Word Processing) Command

The Start Word Processing (STRWP) command shows a menu of office word processing functions.

More information on word processing is in the *Word Processing User's Guide.*



OPTION Parameter

Specifies the option selected from the menu.

__*SELECT:__ The Word Processing menu is shown to the user so that the user can select an option from the menu.

main-menu-options: Specify the option number selected from the menu. The valid options are as follows:

1. Work with documents in folder
2. Print or view status of document
3. Work with folders
4. Work with non-text document data
5. Work with text profiles

Example

```
STRWP OPTION(*SELECT)
```

This command shows the Word Processing menu.

TFRBCHJOB (Transfer Batch Job) Command

The Transfer Batch Job (TFRBCHJOB) command transfers a batch job to the specified job queue. The job queue does not have to be allocated to an active subsystem at the time of the batch job transfer. The batch job that is transferred is the one in which this TFRBCHJOB command is issued. Routing data and request data can be specified for the batch job when it is transferred. The routing data specified is processed in the subsystem in which the job queue is active. The request data is placed after all other request data for the job. The transferred batch job will resume running at the request data immediately following the TFRBCHJOB command. If the TFRBCHJOB command is issued in a CL program, all subsequent commands in the CL program are bypassed. If objects that are allocated to the current routing step or files that are open in the current routing step are needed in the new routing step, they must be allocated or opened again after the new routing step has been started.

A batch job transferred to a job queue by the TFRBCHJOB command will exist through an initial program load (IPL) if the batch job was residing on the job queue at the time the system was powered down; the job's temporary objects exist through a power down. A batch job's temporary objects are destroyed during the power down if the job was transferred by the Transfer Job (TFRJOB) command.

The QTEMP library of a batch job that has been transferred by the TFRBCHJOB command is always empty when the next routing step is started. Caution must be used with the library list in conjunction with a batch job that was transferred to a job queue by the TFRBCHJOB command. The TFRBCHJOB function saves the library list to recover the job on a job queue if an IPL occurs. When the routing step for the transferred batch job is started, the libraries in the saved library list must exist in the system or the job's routing step ends.

Restrictions: You must have add and read authority for the job queue and operational authority for the subsystem to which the job queue is allocated. The job being transferred must be a batch job. The TFRBCHJOB command is not allowed to run in a batch communications job (a batch job that was started as the result of a program start request). Also, interactive, auto-start, reader, writer, and system jobs will not be allowed to be transferred by the TFRBCHJOB command.

```
                                                          ┌─┐
                       ┌►*LIBL/─────────┐                 │P│
TFRBCHJOB──JOBQ──┼─*CURLIB/───────┼──job-queue-name──┴─┴──────────────►
                       └─library-name/──┘

                                                                   Required
─────────────────────────────────────────────────────────────────────────────
                                                                   Optional
             ┌►QCMDB──────────┐             ┌►*NONE──────────┐
►──RTGDTA──┼─*RQSDTA────────┼───RQSDTA──┼─*RTGDTA────────┼─────────────────
             └─'routing-data'─┘             └─'request-data'─┘
                                                       ┌─────────────────────┐
                                                       │ Job: B  Pgm: B  Exec │
                                                       └─────────────────────┘
```

JOBQ Parameter

Specifies the qualified name of the job queue where the batch job is transferred.

The possible library values are:

***LIBL:** The library list is used to locate the queue.

***CURLIB:** The current library for the job is used to locate the queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the queue is located.

RTGDTA Parameter

Specifies the routing data used with this job description to start jobs. The routing data is used to determine the routing entry (in the subsystem description) that identifies the program where the job runs.

QCMDB: This routing data matches a routing entry in the IBM-supplied subsystem description, which starts a routing step processed by the IBM-supplied control language processor QCL.

***RQSDTA:** Up to 80 characters of the request data specified in the RQSDTA parameter of this command are used as the routing data for the routing step. If this value is used, RQSDTA(*RTGDTA or *NONE) cannot be specified.

'routing-data': Specify the character string that is used as the routing data for starting the routing step. A maximum of 80 characters can be entered (enclosed in apostrophes if blanks or special characters are included).

RQSDTA Parameter
Specifies the request data that is added to the end of the job's message queue for use by the new routing step.

***NONE:** No request data is placed in the job's message queue.

***RTGDTA:** The routing data specified in the RTGDTA parameter is placed at the end of the job's message queue. If this value is used, RTGDTA(*RQSDTA) cannot be specified.

'request-data': Specify the character string that is placed at the end of the job's message queue for use by the new routing step or some subsequent routing step in the job. A maximum of 256 characters can be entered (enclosed in apostrophes if blanks or special characters are included). When a CL command is entered, it must be enclosed in single apostrophes, and where apostrophes would normally be used in the command, double apostrophes must be used.

Example

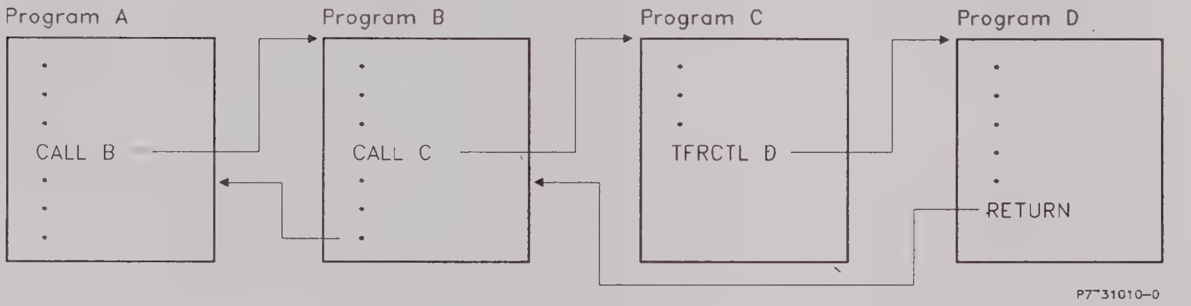
```
TFRBCHJOB  JOBQ(QGPL/QCTL)  RTGDTA(APPLICS)
```

This command transfers the batch job where the command is entered to the QCTL job queue that is in the QGPL library. The job is routed using the routing data APPLICS. The job must be a batch job.

TFRCTL (Transfer Control) Command

The Transfer Control (TFRCTL) command calls the program specified on the command, passes control to it, and removes the transferring program from the return stack. With the transferring program removed from the program stack, control does not return to it when the called program returns control. Instead, control is returned to the command following the last call to the transferring program.

The following diagram shows the operation of both the CALL and TFRCTL commands. First, Program A calls Program B. Program B then calls Program C, which in turn transfers control (using the TFRCTL command) to Program D. In transferring control to Program D, Program C is removed from the program stack; therefore, Program D returns to Program B (to the command following the most recent call to the transferring program, Program C). Finally, Program B returns to Program A.



If the transferring program was created with USRPRF(*OWNER), the authority does not transfer. However, if a program created with USRPRF(*OWNER) calls a program that transfers control, the authority *does* transfer because the program that called the transferring program remains in the program stack.

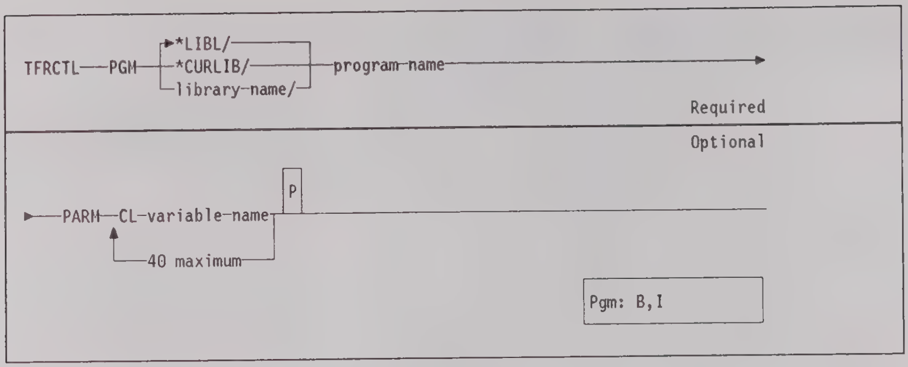
Optionally, the transferring program can pass parameters to the program being called. The storage space used by the CL variables in the transferring program is freed and made available for use by the program receiving control.

The parameter values must be passed in CL variables. Values cannot be passed as constants, as null parameters (that is, parameters whose values are null, specified by *N), as lists of values, or as CL variables that were not specified as parameters on the PGM command that identified the start of the transferring program.

The transferring program can only pass CL variables that were previously passed to it. No CL variable can be passed that exists within the transferring program itself. A maximum of 40 parameters can be passed to the called program. The parameters passed must agree in type, length, number, and order with those expected by the receiving program, as specified in the PARM parameter of the PGM command.

Shared files remain open if control is transferred to a program from a high-level language program or from the CL program that opened the files.

Restrictions: (1) This command is valid only within CL programs. (2) The user must have *CHANGE authority or one of the data authorities for the program to which control is being transferred.



PGM Parameter

Specifies the qualified name of the program that receives control from the program transferring control.

The possible library values are:

***LIBL:** The library list is used to locate the program.

***CURLIB:** The current library for the job is used to locate the program. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the program is located.

PARM Parameter

Specifies the names of one or more CL variables that are passed to the program receiving control. The variables passed can only be parameters that were themselves passed to the program currently transferring control. A maximum of 40 variables can be specified. Specify a CL variable name for each of the values to be passed by the program transferring control. The parameter values are received in the receiving program in the order in which they were specified on the TFRCTL command that calls the program.

The value *N cannot be specified as a value being passed because a null value cannot be passed to another program.

Example

```
TFRCTL   PROGA  &PARM1
```

This command transfers control to the program PROGA and passes the parameter &PARM1 to it. The parameter &PARM1 must previously have been passed to the program issuing this command.

TFRGRPJOB (Transfer to Group Job) Command

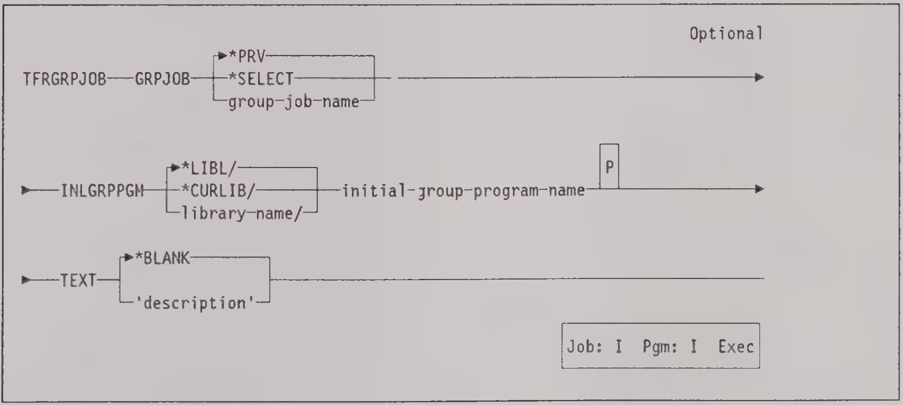
The Transfer to Group Job (TFRGRPJOB) command suspends the job that issued the TFRGRPJOB command and the group job specified by the GRPJOB parameter is resumed (if it already exists) or is created (if it does not exist). In both cases, control is transferred to the job specified by the GRPJOB parameter.

The job issuing the TFRGRPJOB command remains suspended until control is passed back to it and the job is resumed. A suspended job is not able to do any processing until it is resumed. Suspended group jobs can regain control if another group job issues the TFRGRPJOB command to transfer to the suspended job, or if the active job in the group ends (and the suspended job was specified as the job to gain control).

If the work station message queue is in break or notify mode in the job that issued the TFRGRPJOB command, the message queue is set to the same mode in the group job that is transferred to. For example, if group job A has the work station message queue in break mode and transfers to group job B, the queue is in break mode in group job B. If group job B changes the queue to notify mode and transfers back to group job A, the queue is in notify mode in group job A.

If a user message queue is associated with the group (Change Group Attributes (CHGGRPA) command), it is handled like the work station message queue when one group job transfers to another group job.

Note: No message is sent for normal completion of the TFRGRPJOB command. Instead, you may obtain a control code describing the circumstances under which the job regained control (along with the group job name and the job number of the previously active job) by using the RTVGRPA command. For more information about these control codes, see the Retrieve Group Attributes (RTVGRPA) command.



GRPJOB Parameter

Specifies the group job to be transferred to.

***PRV:** The previously active job in the group is transferred to. If the previously active job no longer exists, then the most recently active job in the group is resumed. This special value is valid only if there is another group job in the group.

***SELECT:** The group job selection display is shown. You can choose which group job to transfer to or create a new group job and transfer to it.

group-job-name: Specify the group job name of the job that is transferred to.

When a group job name is specified and that group job already exists, the group job is resumed. Whenever an already existing group job is resumed, the INLGRPPGM and TEXT parameters are ignored. However, when a group job name is specified and that group job does not already exist, the INLGRPPGM parameter is required and must specify an existing program. When this is the case, a group job is started and control is passed to the new group job.

INLGRPPGM Parameter

Specifies the name of the job's first group program. The INLGRPPGM parameter only has meaning when a group job is created. If the group job being transferred to already exists, this parameter is ignored.

The possible library values are:

*LIBL: The library list is used to locate the program.

***CURLIB:** The current library for the job is used to locate the program. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the program is located.

TEXT Parameter

Specifies up to 50 characters of text that describes the group job. This parameter only has meaning when a group job is created. If the group job being transferred to already exists, this parameter is ignored.

*BLANK: The text is all blanks.

'description': Specify no more than 50 characters, enclosed in apostrophes.

Note: This text appears on the Group Job Selection.

Example

```
TFRGRPJOB  GRPJOB(GROUPJ1)  INLGRPPGM(QGPL/PROGRAM1)
```

This command suspends running of the current job. If group job GROUPJ1 already exists, it is resumed at the point where it was suspended (the next high-level language command following the TFRGRPJOB request).

If group job GROUPJ1 does not exist, then a group job is created; it is called GROUPJ1 and runs program QGPL/PROGRAM1.

TFRJOB (Transfer Job) Command

The Transfer Job (TFRJOB) command transfers a job to the specified job queue that is run in the subsystem where that queue is active. The job that is transferred is the one where this TFRJOB command is issued. The specified job queue is normally in a different subsystem than the one the job is currently in. If the job being transferred is an interactive job, it is given the highest priority on the job queue. New routing data and request data can be specified for the job when it is transferred. If objects that were allocated to the previous routing step or files that were open in the previous routing step are needed in the new routing step, they must be allocated or opened again.

Note: Running this command in a batch job causes loss of spooled inline files because they cannot be accessed in the new routing step.

Also, if the target subsystem is ended (by running the End Subsystem (ENDSBS) command, the End System (ENDSYS) command, or the Power Down System (PWRDWNSYS) command) while an interactive transferring job is on a job queue, the job is canceled as part of subsystem ending.

Because a PWRDWNSYS command inhibits new jobs or routing steps from being started by any subsystem, a batch job transferred to a job queue (by the TFRJOB command) is not completed before the system is powered down. The temporary objects associated with a transferring job (such as the library list, the QTEMP library, and all objects in it) are destroyed during PWRDWNSYS, so that during a re-IPL, the system is unable to restore the job to its previous state. During re-IPL, the system removes the job from the job queue and produces its job log.

Restrictions: The user must have read and add authority for the job queue and for the subsystem that the job queue is allocated to. If the job being transferred is an interactive job, the following restrictions apply:

- The job queue where the job is placed must be associated with an active subsystem.
- The work station associated with the job must have a corresponding work station entry in the subsystem description associated with the new subsystem.
- The work station associated with the job must not have another job associated with it that has been suspended by means of the Sys Req (system request) key. The suspended job must be canceled before the Transfer Job command can be run.
- The job must not be a group job.
- The job must not be a communications batch job (started as a result of a program start request), unless it meets one of the following criteria:
  - It was started from an APPC communications device, or
  - The session on the communications device hase ended.

```
                                                   ┌─┐
                ┌►*LIBL/─────────┐                 │P│
TFRJOB──JOBQ────┼─*CURLIB/───────┼─job-queue-name──┴─┴──────────────────────►
                └─library-name/──┘
                                                                    Required
─────────────────────────────────────────────────────────────────────────────
                                                                    Optional
              ┌►QCMDI──────────┐              ┌►*NONE─────────┐
►──RTGDTA─────┼─*RQSDTA────────┼──RQSDTA──────┼─*RTGDTA───────┼──────────────
              └─'routing-data'─┘              └─'request-data'┘
                                                   ┌─────────────────────────┐
                                                   │Job: B,I  Pgm: B,I  Exec │
                                                   └─────────────────────────┘
```

JOBQ Parameter
Specifies the qualified name of the job queue where the job is transferred to.

The possible library values are:

***LIBL:** The library list is used to locate the queue.

***CURLIB:** The current library for the job is used to locate the queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the queue is located.

RTGDTA Parameter

Specifies the routing data used with this job description to start jobs. The routing data is used to determine the routing entry (in the subsystem description) that identifies the program where the job runs.

QCMDI: This routing data matches a routing entry in the IBM-supplied QINTER subsystem description, which starts a routing step processed by the IBM-supplied control language processor QCL, in the QSYS library.

*RQSDTA: Up to 80 characters of the request data specified in the RQSDTA parameter of this command are also used as the routing data for the routing step.

'routing-data': Specify the character string that is used as the routing data for starting the routing step. A maximum of 80 characters can be entered (enclosed in apostrophes if necessary).

RQSDTA Parameter

Specifies the request data that is added to the end of the job's message queue for use by the new routing step.

***NONE:** No request data is placed in the job's message queue.

***RTGDTA:** The routing data specified in the RTGDTA is also placed at the end of the job's message queue.

'request-data': Specify the character string that is placed at the end of the job's message queue for use by the new routing step or some subsequent routing step in the job. A maximum of 256 characters can be entered (enclosed in apostrophes if necessary). When a CL command is entered, it must be enclosed in single apostrophes, and where apostrophes would normally be used *inside* the command, double apostrophes must be used instead.

Example

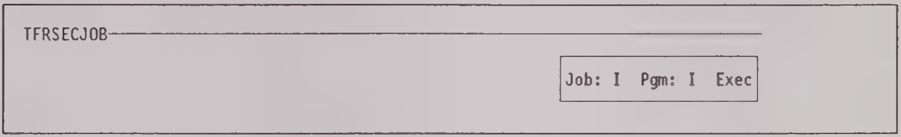
```
TFRJOB  JOBQ(QGPL/QCTL)  RTGDTA(APPLICS)
```

This command transfers the job where the command is entered to the QCTL job queue in the QGPL library. The job is routed using the routing data APPLICS. If the job is an interactive job, the job queue must be allocated by an active subsystem.

TFRSECJOB (Transfer Secondary Job) Command

The Transfer Secondary Job (TFRSECJOB) command creates a secondary interactive job at your work station, then transfers control between the primary and secondary jobs. The first time you issue this command, you receive the signon prompt for the secondary job. Once you sign on, a secondary job is created, allowing you to receive the basic working display of the new job (such as the command entry display). Your primary job remains suspended as long as you remain in your secondary job. The next time you issue the TFRSECJOB command, your current job is suspended, and you return to the first job at the point at which you left it. When you sign off either job, you are automatically returned to the remaining job.

Note: This command performs the same function as option 1 on the System Request Menu.



There are no parameters for this command.

Example

```
TFRSECJOB
```

The job that is currently running is suspended, and the SIGNON prompt is displayed. If a secondary job already exists, that job resumes (the SIGNON prompt is not displayed).

TRCINT (Trace Internal) Command

The Trace Internal (TRCINT) command is used primarily for problem analysis. It controls traces of internal events associated with the current job that occur at a level below the machine interface. Specific types of traces can be started and stopped using this command. Those that apply to devices can be limited to a particular device.

While previously started internal traces are being performed, additional internal traces can be started through this command. The output created by the trace is placed in internal storage used by the internal trace command. The records from the internal storage can be written to a spooled printer file or written to a diskette.

```
                   ┌*ON───┐
                   ├*OFF ─┤
TRCINT──SET────────┼*END──┼────────────────────────────────────────────────►
                   ├*HOLD─┤
                   ├*SAVE─┤
                   └*SIZE─┘
                                                                   Required
───────────────────────────────────────────────────────────────────────────
                                                                   Optional
                                     ┌►*NONE──────────┐
►──TRCTYPE──trace-type────DEV──┤                │────────────────────────►
              ▲           │         └device-name───┤
              └50 maximum─┘           ▲           │
                                      └16 maximum─┘

      (2)         ┌►*WRAP───────┐   ┌─┐      (1)
►──TRCFULL────────┤             ├───┤P├──────OUTDEV device-name────────────►
                  └*STOPTRC─────┘   └─┘

      (3)     ┌►*MAX──────────┐
►──SIZE───────┼─*MIN──────────┼───────────────────────────────────────────
              └─table-size────┘

(1) OUTDEV only used with SET(*SAVE).
(2) TRCFULL can only be specified with SET(*END).
(3) SIZE is only used with SET(*SIZE).

                                              ┌───────────────────────────┐
                                              │Job: B,I  Pgm: B,I   Exec  │
                                              └───────────────────────────┘
```

SET Parameter

Specifies whether internal tracing is started, stopped, ended, held or saved. Also, you can specify whether the trace table size is changed.

***ON:** The collection of internal trace records is to be started for the trace types specified in the TRCTYPE parameter. If the trace file already contains trace records, the new trace records are added to the file. If the file becomes full, the action specified in the TRCFULL parameter is taken.

***OFF:** All collection of internal trace records requested through previous TRCINT commands stops, and the records are written to the spooled printer file QPCSMPRT.

***END:** All internal tracing ends. All internal trace records are destroyed.

***HOLD:** All internal traces are stopped, and the internal trace records that were collected are held in internal storage. Held records can be printed later if another TRCINT command is entered that specifies SET(*OFF), or they can be put on tape or diskette if SET(*SAVE) is specified.

***SAVE:** All internal traces are stopped, and the internal trace records are written to the tape or diskette device specified by the OUTDEV parameter. The diskettes that the records are written on must be initialized in the basic exchange format (*DATA or *DATA2).

***SIZE:** The size of the trace collection table is changed. The new size to use is specified in the SIZE parameter.

TRCTYPE Parameter

Specifies the type or types of traces to start. If any types are specified, SET(*ON) must also be specified. If any value other than *ON is specified for the SET parameter, TRCTYPE is ignored. Each trace type is identified by a 6-digit code; all 6 digits must be specified. Specify up to 50 of the following types:

Code	Type of Trace
030000	System/38 instruction supervisor linkage (SVL) trace
040000	Multiprogramming level (MPL) trace
060000	Task switch trace

Code	Component Data Trace
010100	Resource management
010200	Exception management
010300	Program management
010400	Data base management (events for all data base files are traced)
010500	Process management
010600	Event management
010700	Queue management
010801	Load/dump (save/restore)
010802	Machine services control point
010803	Source/sink device (device support)
010804	Display station pass-through
010900	Authority management
011000	Context management
011101	Auxiliary storage management
011102	Main storage management – invocations
011103	Main storage management – details
011104	Storage management – ERP
011105	All storage management
011200	Common functions
011300	Machine observation
011400	Independent index management
011500	Space object management
011600	Journal management
011700	Commit management
011900	Common class input/ouput manager (CCIOM)
012000	Inter-process communications facility (IPCF)
012100	Error log
012200	Environmental recording, editing and printing (EREP)
012300	Communication trace service function
012400	Link test service function
012501	APPN control point manager
012502	APPN directory services
012503	APPN topology and routing services
012504	APPN control point presentation services
012505	APPN location manager
012506	All APPN traces
050100	Common functions
050200	Authority management
050300	Commit management
050400	Context management
050500	Data base management
050600	Exception management
050700	Event management
050800	Independent index management
050900	Journal management
051000	Process management
051100	Program management
051200	Resource management
051300	Source/sink (device support) management
051400	Space object management
051500	Machine services control point
051600	Machine observation
051700	Recovery management
051800	Queue management
051900	Source/sink IOM
052000	Load/dump
052100	Common class input/output manager (CCIOM)
052200	Inter-process communications facility (IPCF)
052300	Error log
052400	Environmental recording, editing and printing (EREP)
052500	Communication trace service function
052600	Link test service function
052700	APPN control point manager
052800	APPN directory services
052900	APPN topology and routing services
053000	APPN control point presentation services
053100	APPN location manager

DEV Parameter

Specifies the names of the devices that the associated internal events are traced for. This parameter can only be specified if TRCTYPE(010803) is specified; otherwise, it is ignored.

***NONE:** No devices are traced by this command.

device-name: Specify the names of up to 16 devices that the internal trace is started for. The device names must be the same as the names specified in the associated device descriptions.

TRCFULL Parameter

Specifies whether the trace records wrap (replace the oldest records with new records) or stop tracing when all of the storage is used. This parameter may only be specified when SET(*END) is used.

***WRAP:** When the trace storage is full, the trace wraps to the beginning. The oldest trace records are written over by new ones as they are collected.

***STOPTRC:** Tracing is stopped when the trace storage is full of trace records.

OUTDEV Parameter

Specifies the device description name of the tape or diskette device that the held trace records are to be written on. This parameter must be specified if SET(*SAVE) is specified, otherwise it is ignored.

SIZE Parameter

Specifies the size of the data collection table. This parameter may only be specified when SET(*SIZE) is used.

***MAX:** The trace recording table is set to maximum size of 16000K bytes.

***MIN:** The trace recording table is set to the minimum size of 2K bytes.

table-size: Specify the size that the trace recording table should be (specify an integer between 2 and 16000).

Examples

```
TRCINT   SET(*ON)  TRCTYPE(010100 010400 050500 051200)
```

This command starts component data traces and call traces for resource management and data base. All data base operations associated with all data base files are used to collect component data trace records.

```
TRCINT   SET(*ON)  TRCTYPE(010803)  DEV(WS1 WS2 WS3)
```

This command starts component data traces for source/sink management (device support) operations involving the devices WS1, WS2, and WS3.

```
TRCINT   SET(*END)  TRCFULL(*STOPTRC)
```

This command stops all traces and clears the trace storage. When TRCINT is used again, tracing stops when the trace storage becomes full.

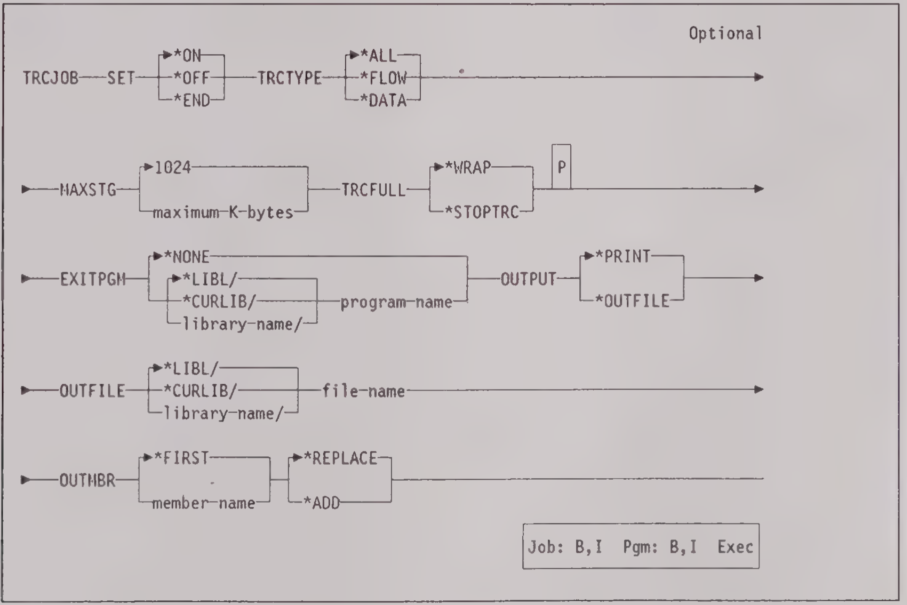
TRCJOB (Trace Job) Command

The Trace Job (TRCJOB) command controls traces of program calls and returns that occur in the current job or in the job being serviced as a result of the Start Service Job (STRSRVJOB) command directed to that job. The command, which sets a trace on or off, can trace module flow, OS/400 data acquisition (including CL command traces), or both.

As the trace records are collected, they are stored in an internal trace storage area. When the trace is ended, the trace records can be written to a spooled printer file, QPSRVTRC. The trace records can also be directed to a data base output file.

If the Start Service Job (STRSRVJOB) command is entered before the TRCJOB command, the job that is traced is the one identified by the STRSRVJOB command. The trace output from the serviced job is returned to the servicing job after the trace is set off or after the serviced job has ended.

Restriction: The record format of the data base output file must match the record format of the IBM-supplied output file QATRCJOB.



SET Parameter

Specifies whether the collection of trace records starts or stops.

***ON:** The collection of trace records is started.

***OFF:** The collection of trace records is stopped, and the trace records are written to the spooled printer file.

***END:** The collection of trace records is stopped, and all existing trace records are deleted. No spooled printer file is created.

TRCTYPE Parameter

Specifies the type of trace data to store in a trace file.

***ALL:** All the trace data collected is stored in trace records. This includes tracing the flow of control and the trace data itself.

***FLOW:** The flow of control is traced when programs are called and when they return control.

***DATA:** The data that is provided at predefined trace points within the Operating System/400 is stored in trace records. This includes trace records for the CL commands that have run.

MAXSTG Parameter

Specifies the maximum amount of storage used for collected trace records.

| **1024:** A maximum of 1024 of storage is used.

maximum-K-bytes: Specify the maximum amount of storage, in K-bytes, used to store trace records. (One K equals 1024 bytes.)

TRCFULL Parameter

Specifies whether the trace records are to wrap (replace oldest records with new records) or to set trace off when all of the storage (specified by MAXSTG) has been used.

***WRAP:** When the trace file is full, the trace wraps to the beginning. The oldest trace records are written over by new ones as they are collected.

***STOPTRC:** Tracing stops when the trace file is full of trace records.

EXITPGM Parameter

Specifies the qualified name of a user-written program that is given control just prior to the collection of each trace record.

***NONE:** No user-written program is called.

The possible library values are:

***LIBL:** The library list is used to locate the program.

***CURLIB:** The current library for the job is used to locate the program. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the program is located.

program-name: Specify the name of the user-written program called before each trace record is collected. This program can examine and alter the 78-character trace record passed to it as a parameter. If it replaces the first 2 characters in the trace record with blanks or binary zeros, the entire trace record is suppressed. This program can also call any of the service dump commands and must return control when it completes its task. If the user-written program (which *cannot* be a CL program) runs other OS/400 commands, the output from those commands is associated with the job that they are run in. The dump and trace commands associate their output data either with the job that enters the dump and trace commands or with the job being serviced as the result of the STRSRVJOB command.

OUTPUT Parameter

Specifies if the output from the command is printed with the job's spooled printer output.

***PRINT:** The output is printed with the job's spooled printer output.

***OUTFILE:** The output is put in a data base file. OUTPUT(*OUTFILE) is valid only if a value is specified for the OUTFILE parameter.

OUTFILE Parameter

Specifies the physical data base file that the trace output is directed to. If the output file already exists, the system tries to use it. Records replace existing data in the file member. If the output file does not exist, the system creates a data base physical file (with the name specified in the OUTFILE parameter) in the specified library. A member is created for the file with the name specified in the OUTMBR parameter. If the file is created, the text is "OUTFILE created by TRCJOB command" and *EXCLUDE authority is given to users with no specific authority.

The possible library values are:

***LIBL:** The library list is used to locate the file.

***CURLIB:** The current library for the job is used to locate the file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the file is located.

file-name: Specify the name of the file to receive the trace output.

OUTMBR Parameter

Specifies the name of the member within the file that receives the trace output. If the OUTFILE is created by the system, a member is created for the file with the name specified in the OUTMBR parameter. If the OUTFILE exists but the OUTMBR does not, a member with the specified name is added.

***FIRST:** The first member of the file specified by the OUTFILE parameter receives the trace output. If the OUTFILE is created and *FIRST is specified, the name of the created member is the same as the name of the created file.

member-name: Specify the name of the member, within the file (specified by the OUTFILE parameter) that receives the trace output.

***REPLACE:** The new data is added to the file member in place of the existing data.

***ADD:** The new data is added to the end of the data already in the file member.

Examples

```
TRCJOB   TRCTYPE(*FLOW)  MAXSTG(40)
```

This command traces the module flow of the current job. Trace records are collected for each program call and return that occurs in the job. The trace file contains 40K of storage and wraps if that amount of storage is filled with trace records.

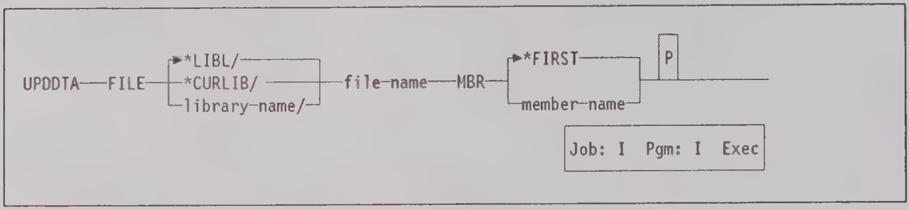
```
TRCJOB   SET(*OFF)  OUTPUT(*NONE)  OUTFILE(QGPL/TRCJOB)  +
  OUTMBR(TRCDTA)
```

This command stops the trace and directs the output to the data base file QGPL/TRCJOB. The output is directed to the member TRCDTA.

UPDDTA (Update Data) Command

The Update Data (UPDDTA) command creates and runs a temporary data file utility (DFU) program. You can use this temporary program to update a data base file.

Restriction: Any overrides on the specified file are ignored.



FILE Parameter
Specifies the qualified name of the data file to be updated.

The possible library values are:

***LIBL:** The library list is used to locate the data file.

***CURLIB:** The current library for the job is used to locate the data file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the data file is located.

MBR Parameter
Specifies the member in the file you want to update.

***FIRST:** You will update the first member of the file.

member-name: Specify the name of the member you want to update.

Example

```
UPDDTA
```

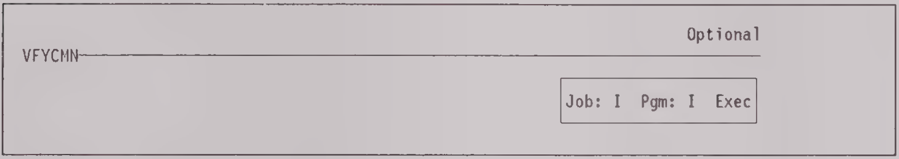
This command creates and runs a temporary DFU program.

VFYCMN (Verify Communications) Command

The Verify Communications (VFYCMN) command shows the Verify a Communications Line display. You can use this command to make sure communications equipment is operating correctly.

Depending on your system's configuration, you will have options to run the following tests:

- Link
- Local modem
- Remote modem
- Cable
- Communications input/output adapter



There are no parameters for this command.

Example

```
VFYCMN
```

This command shows the Verify a Communications Line display.

VFYPRT (Verify Printer) Command

The Verify Printer (VFYPRT) command runs the 3287, 3812 SCS, 3812 IPDS, 4210, 4214, 4224, 4234, 4245, 5219, 5256, 5262, 5524, 5525, 5553, and 5583 Printers by causing them to print a test pattern a specified number of times.

```
                          ┌─┐
                          │P│
VFYPRT───DEV device-name──┴─┴──────────────────────────────────────────►
                                                               Required
───────────────────────────────────────────────────────────────────────
                                                               Optional
►───TIMES──┬►1──────────────┬──────────────────────────────────────────
           └─number-of-times┘

                                            ┌────────────────────────┐
                                            │Job: B,I  Pgm: B,I  Exec│
                                            └────────────────────────┘
```

DEV Parameter
Specifies the name of the printer to run the test pattern on. The device name must be the same as that specified in the device description for the printer.

TIMES Parameter
Specifies the number of times that the specified printer prints the test pattern.

1: The test pattern is printed one time.

number-of-times: Specify a value for the number of times to print the test pattern.

Example

```
VFYPRT   DEV(PRTR3)  TIMES(15)
```

This command causes the printer named PRTR3 to print a test pattern 15 times.

VRYCFG (Vary Configuration) Command

The Vary Configuration (VRYCFG) command varies configuration objects on or off and optionally resets the input/output processor (IOP) associated with the specified objects. The VRYCFG command is used to vary on or off one or more configuration objects with the capability of also varying on the downline attached configuration objects. The configuration object types that can be varied are line, controller, and device. This command applies to all lines, controllers, and devices on the system.

Downline attached objects can be varied on or off along with the specified object by specifying the value NET for the range parameter (RANGE(*NET)). Downline attached objects of a line are all the attached controllers and all the devices attached to the controllers. Downline attached objects of a controller are all the attached devices. Devices do not have downline attachments. The range parameter has no affect when varying devices.

Varying on lines synchronously or asynchronously can be controlled by the VRYWAIT parameter. This applies only to token-ring, X.25, or switched SDLC, BSC, or Async line descriptions. The value specified for the VRYWAIT parameter determines how long the system waits until either the line goes to varied on before completing the VRYCFG command, or until the timer expires.

The VRYCFG command can also be used to reset the IOP. An IOP can be a communications controller, local work station, or magnetic media controller. An IOP reset is valid only when varying on lines (except twinaxial data link controller (TDLC) lines), local work station controllers, tapes and diskettes.

A controller cannot be varied on:

- If the line to which it is attached is varied off, in the case of leased (nonswitched) lines.
- Until a dial connection has been completed, in the case of switched lines.

A device cannot be varied on:

- If the controller to which it is attached is varied off. In the case of tape and diskette devices, they are not attached to a controller, so this restriction does not apply.

A line cannot be varied off:

- Until all the attached controllers and devices are varied off.

A controller cannot be varied off:

- If it is being used or allocated for use.
- Until all the attached devices are varied off.

A device cannot be varied off:

- If it is being used or allocated for use.

Note: Vary Wait parameter (VRYWAIT) use:

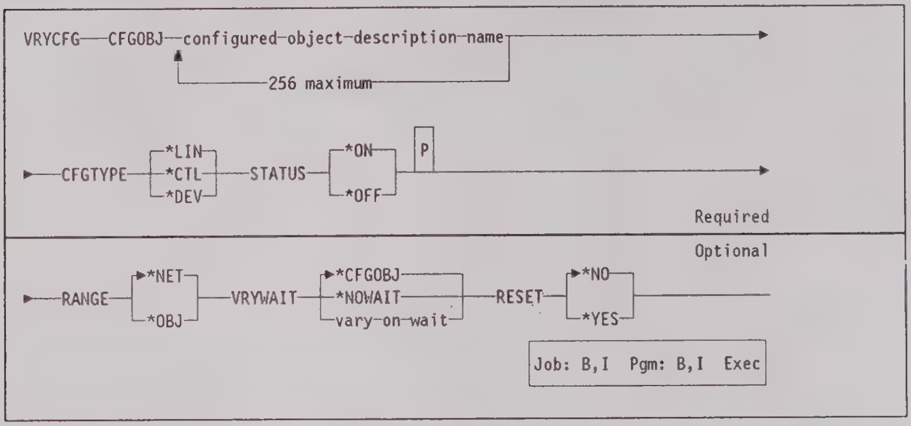
The time to vary on a line is the time it takes to put tasks in place to manage the line, the time to activate the communications I O processor (IOP), including download of the IOP microcode, the time to establish communications with the data circuit-terminating equipment (DCE) and other unique protocol set up tasks.

Line vary on time does not include telephone dialing time, however; a powered off modem may prevent vary on completion and cause the wait time to expire. If the timer expires, an informational message is sent to the QSYSOPR message queue. This is followed by the vary completion message.

If the VRYWAIT parameter is specified on the VRYCFG command for a line description that is not token-ring, X.25, or switched SDLC, BSC, or Async, the parameter is accepted but ignored.

Note: Range parameter (RANGE) use:

- For devices: The value *NET to vary on or off downline attached objects has no affect. Devices do not have downline attached objects.
- For switched lines: The value *NET, only when varying on, has no affect. The value *NET, when varying off, will vary off the line and its downline attached objects.
- Default value is *NET.



CFGOBJ Parameter
Specifies the name of the configuration object to be varied on or off. The name can have a maximum of 10 characters.

CFGTYPE Parameter
Specifies the type of object to be varied.

***LIN:** The line is varied.

***CTL:** The controller is varied.

***DEV:** The device is varied.

STATUS Parameter
Specifies if the system is to vary the object on or off.

***ON:** The object is varied on.

***OFF:** The object is varied off.

RANGE Parameter
Specifies what configuration elements are varied.

***NET:** All downline attached configuration elements are varied.

***OBJ:** Only the specified objects are varied.

VRYWAIT Parameter
Specifies whether the token-ring, X.25, or switched SDLC, BSC, or Async line description is varied on asynchronously or synchronously. Specify a wait time (synchronously vary on) when an application will open/acquire to a communications file immediately after issuing the vary on of the communication description.

***CFGOBJ:** Uses the VRYWAIT parameter value specified in the line description.

***NOWAIT:** Do not wait for vary on completion. The line is varied on asynchronously.

vary-on-wait: Specify a value from 15 to 180 seconds in 1 second intervals. The system waits until either the line goes to varied on before completing the VRYCFG command or the timer expires.

RESET Parameter
Specifies if a reset is to be done for the IOP associated with the object.

***NO:** The associated IOP is not reset.

***YES:** The associated IOP is reset.

Examples

```
VRYCFG  CFGOBJ(LINE1)  OJBTYPE(*LIN)  STATUS(*OFF)
```

This command varies off the line and all attached downline objects. The range parameter (RANGE) took the default value of *NET.

```
VRYCFG  CFGOBJ(CONTROLLER1)  OBJTYPE(*CTL)  STATUS(*ON)  +
  RANGE(*OBJ)
```

This command varies on only the controller.

```
VRYCFG  CFGOBJ(DEVICE1)  OBJTYPE(*DEV)  STATUS(*ON)  RANGE(*NET)
```

This command varies on only the device. Note the RANGE value has no affect on devices.

```
VRYCFG  CFGOBJ(LINE1)  OBJTYPE(*LIN)  STATUS(*ON)  +
  RANGE(*OBJ)  RESET(*YES)
```

This command will vary on only the line and reset the associated IOP.

```
VRYCFG  CFGOBJ(LINE1)  OJBTYPE(*LIN)  STATUS(*ON)  +
  RANGE(*OBJ)  VRYWAIT(*CFGOBJ)
```

This command varies on only the line and uses the vary wait time value specified in the line description for LINE1.

```
VRYCFG  CFGOBJ(LINE1)  OBJTYPE(*LIN)  STATUS(*ON)  +
  RANGE(*OBJ)  VRYWAIT(80)
```

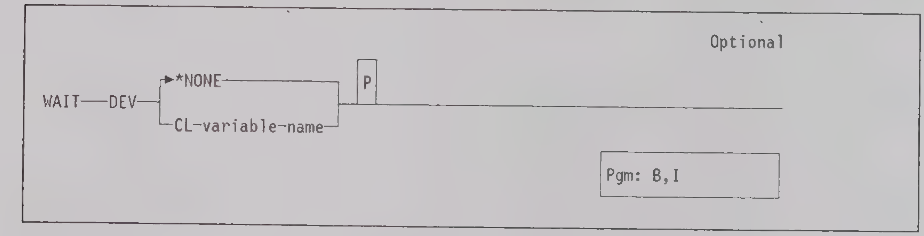
This command varies on only the line using 80 seconds as the vary wait time value.

WAIT (Wait) Command

The Wait (WAIT) command accepts input from any display device from which user data was invited by one or more previous Receive File (RCVF), Send File (SNDF), or Send/Receive File (SNDRCVF) commands that did not want to wait for the input data to be received. Those commands had *NO specified in the WAIT parameter or, in the case of SNDF, had the INVITE DDS keyword optioned on in the record format sent to the display, and specified a particular device file to receive and transfer the data to the CL program. Only one input request *per device* can be outstanding at any given time. If there are multiple outstanding input requests, the first device to respond to the specified device file has its user data sent to the CL program. If the data is received within the wait interval, the Wait operation ends and the next command in the program is processed. Otherwise an escape message is sent to the CL program.

The program waits for the number of seconds specified on the WAITRCD keyword of the Create Display File (CRTDSPF), Change Display File (CHGDSPF), or Override with Display File (OVRDSPF) commands for a device to respond to an input request.

Restriction: This command is valid only within a CL program. This command is valid only for display files. It cannot be used with data base files.



DEV Parameter
Specifies the name of the CL variable that receives the name of the display device that responds with user data for the CL program.

NONE: No CL variable name is specified; the name of the responding device is not needed.

CL-variable-name: Specify the name of the CL variable that receives the name of the responding device. The variable must be a character variable with a maximum length of 10. A device name cannot be specified in this parameter.

Examples

```
DCLF   FILE(MSCREEN)
  •
  •
  •
RCVF   DEV(DEV1)  WAIT(*NO)
  •
  •
  •
RCVF   DEV(DEV2)  WAIT(*NO)
  •
  •
  •
WAIT  DEV(&DEVNAM)
```

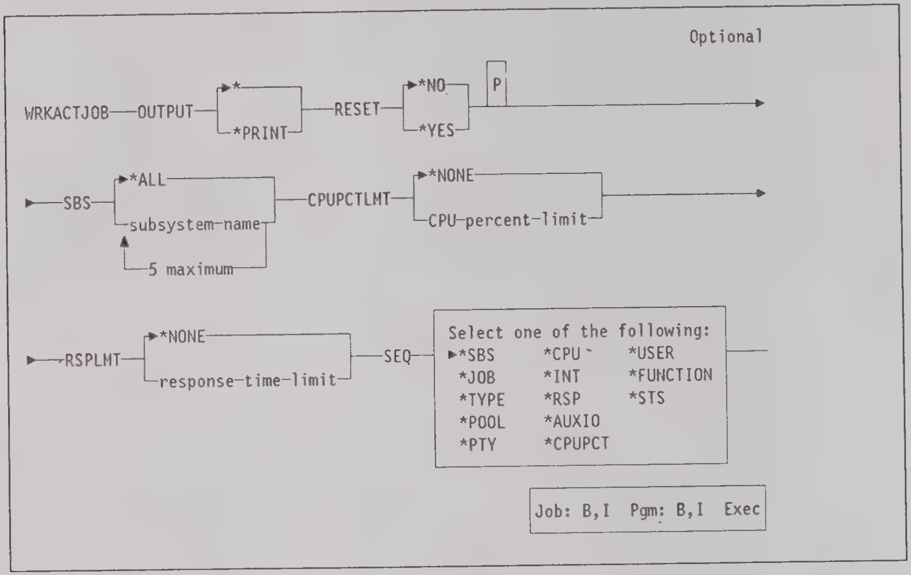
In this example, the device file MSCREEN is used to receive user data; the RCVF commands specify that the program does not wait for the data. The WAIT command causes the program to wait for the display device file MSCREEN to pass input data to it from one of its devices. The name of the responding display device is placed into the CL variable &DEVNAM. The received data is placed in the program variables associated with the record format of the declared file.

```
DCLF   FILE(DF1)
  •
  •
  •
RCVF   DEV(DEV1)  WAIT(*NO)
  •
  •
  •
RCVF   DEV(DEV2)  WAIT(*NO)
  •
  •
  •
WAIT   DEV(&DN)
```

In this example, the RCVF commands specify that the program does not wait for user data. When the WAIT command is issued, the name of the responding device is placed in the CL variable &DN. The data received from the device file DF1 is placed in the program variable of the associated record format in the file DF1.

WRKACTJOB (Work with Active Jobs) Command

The Work with Active Jobs (WRKACTJOB) command allows you to work with performance and status information for the active jobs in the system. The SEQ criteria can be changed by operations on the display. Any selection value (CPUPCTLMT, SBS, RSPLMT) cannot be changed by operations on the display.



OUTPUT Parameter

Specifies if the display is shown that allows you to work with or change a job displayed at the requesting work station or printed with the job's spooled output (Refer to Appendix A, "Expanded Parameter Descriptions" for an expanded description of the OUTPUT parameter, and to Appendix D, "Files Used by CL Commands" for the name of the printer file used by this command.)

*: The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

*PRINT: The output is printed with the job's spooled output.

RESET Parameter

Specifies if the active job statistics are reset.

*NO: The active job statistics are not reset. The measurement time interval is extended (similar to pressing F5) if a previous display active jobs command has run in the current job. All active jobs are displayed.

*YES: The active job statistics are reset. A measurement time interval of zero is used (similar to pressing F13). All active jobs are displayed.

SBS Parameter

Specifies the names of the subsystems (or all subsystems) whose active jobs are displayed.

*ALL: The active jobs in all subsystems are displayed. System jobs that are not associated with any subsystem are also displayed.

subsystem-name: Specify the name of the subsystem to be displayed. All active jobs in this subsystem (including the monitor) are displayed. A maximum of five subsystem names may be specified.

CPUPCTLMT Parameter

Specifies the minimum processing time percent value that a job must have before it is included on the display.

*NONE: There is no minimum processing time limit that a job must have to be displayed.

CPU-percent-limit: Specify the minimum processing time percent limit (from 0.1 to 99.9) that a job must have to be included on the display.

RSPLMT Parameter

Specifies the minimum response time limit that a job must have before it is included on the display.

*NONE: There is no minimum response time limit that a job must have to be displayed.

response-time-limit: Specify the minimum response time limit (from 0.1 to 999.9) that a job must have to be included on the display.

SEQ Parameter

Specifies the sequence of the active jobs displayed.

*SBS: The jobs are ordered on the basis of what subsystem they are running in. Jobs that run in a subsystem (auto-start jobs, interactive jobs, batch jobs, readers, and writers) are put in alphabetical order by job name, and are indented under the subsystem they are associated with. Subsystem monitors (with the jobs in the subsystem grouped under each monitor job) are put in alphabetical order and presented before system (SYS) jobs. The system jobs (start operating system, system arbiter, and LU services) are put in alphabetical order by job name, and are presented after the subsystem monitors and jobs in the subsystems.

***JOB:** Jobs are put in alphabetical order by job name.

***TYPE:** Jobs are put in alphabetical order by job type and job name within the same type.

***POOL:** Jobs are ordered by the system pool they are running in. The lowest values are presented first.

***PTY:** Jobs are ordered by priority of running. The highest priority values (0) are presented first.

***CPU:** Jobs are ordered by the amount of processing time they have used since the job started. The largest values are presented first.

***INT:** Jobs are ordered by the number of operator interactions that have occurred during the measurement interval. The largest values are presented first. Jobs that have no interactions (blank interaction field) come after jobs with 0 interactions.

***RSP:** Jobs are ordered by the average response time during the measurement interval. The largest values are presented first. Jobs that have no interactions (blank interaction field) come after jobs with 0 response time.

***AUXIO:** Jobs are ordered by the number of auxiliary storage input/output (I/O) operations that have occurred during the measurement time interval. The largest values are presented first.

***CPUPCT:** Jobs are ordered by the percent of CPU resource they have used during the measurement interval. The largest values are presented first.

***USER:** Jobs are ordered alphabetically by user name.

***FUNCTION:** Jobs are put in alphabetical order by the contents of the function field.

***STS:** Jobs are put in alphabetical order by the contents of the status field.

Examples

```
WRKACTJOB  RESET(*YES)  CPUPCTLMT(2)
```

This command presents a display with no jobs appearing; the active job statistics are reset and no job has used any processing unit time since the reset point. When the display appears, F5 may be pressed; this causes a display of all jobs which have exceeded 2 percent of the processing unit utilization since the reset point.

```
WRKACTJOB  SBS(QINTER)  SEQ(*INT)
```

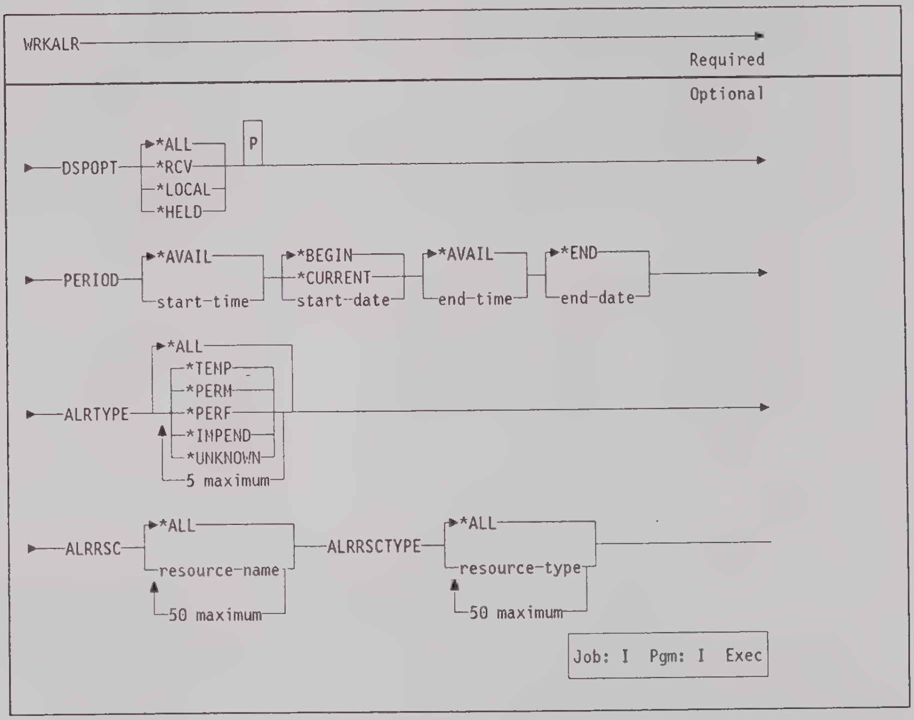
This command allows you to work with all jobs in the QINTER subsystem. The sequence of the jobs is by the number of operator interactions, with the job with the most interactions appearing first.

WRKALR (Work with Alerts) Command

The Work with Alerts (WRKALR) command shows alerts that were created by your system or received from another system.

More information on alerts is in the *C & SM User's Guide.*

Restriction: This command is valid only in an interactive environment.



DSPOPT Parameter

Specifies if alerts received from other systems and/or created locally are shown. Alerts that cannot be sent to the system focal point and are marked held can be shown.

***ALL:** All alerts that are received and locally created, are shown.

*RCV: Only alerts received from other systems are shown.

*LOCAL: Only locally created alerts are shown.

*HELD: All alerts that cannot be sent to the system's focal point and are marked HELD are shown.

Note: There is a distinction between held alerts that are sent or forwarded by this system, and held alerts that are received by another system. DSPOPT(*HELD) displays only held alerts that could not be sent or forwarded by this system.

PERIOD Parameter

Specifies the period of time for which the logged data is displayed. The following values can be coded in this parameter, which contains two lists of two elements each. If PERIOD is not specified, the following values are assumed:

```
PERIOD((*AVAIL *BEGIN)
    (*AVAIL *END))
```

Starting Time: One of the following is used to specify the starting time at which or after which the data must have been logged. Any entries logged before the specified time and date are not displayed.

***AVAIL:** The logged data that is available for the specified starting *date* is displayed.

start-time: Specify the starting time for the specified starting date that indicates the logged data that is displayed. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds. If colons are used to separate the time values, the string must be enclosed in apostrophes ('hh:mm:ss').

Starting Date: One of the following is used to specify the starting date on which or after which the data must have been logged. Any entries logged before the specified date are not displayed.

***BEGIN:** The logged data from the beginning of the log is displayed.

Note: If *BEGIN is specified, then any time value other than *AVAIL for start-time is ignored.

***CURRENT:** The logged data for the current day and between the specified starting and ending times (if specified) is displayed.

start-date: Specify the starting date for which logged data is displayed. The date must be entered in the format specified by the system values QDATFMT and, if separators are used, QDATSEP.

Ending Time: One of the following is used to specify the ending time before which the data must have been logged.

***AVAIL:** The logged data that is available for the specified ending *date* is displayed.

end-time: Specify the ending time for the specified ending date that determines the logged data that is displayed. The time is entered as 4 or 6 digits, and colons can also be used ('hh:mm:ss').

Ending Date: One of the following is used to specify the ending date before which or on which the data must have been logged.

***END:** The last day on which data was logged is the last day for which the logged data is displayed.

Note: If *END is specified, then any time value other than *AVAIL for end-time is ignored.

end-date: Specify the ending date for which logged data is to be displayed. The date must be entered in the format specified by the system values QDATFMT and, if separators are used, QDATSEP.

ALRTYPE Parameter
Specifies which types of alerts are shown. The alert type indicates the severity of the alert.

***ALL:** Shows all types of alerts.

***TEMP:** Shows all alerts reporting a temporary problem.

***PERM:** Shows all alerts reporting a permanent problem.

***PERF:** Shows all alerts reporting a performance problem.

***IMPEND:** Shows all alerts reporting an impending problem.

***UNKNOWN:** Shows all alerts reporting a problem with unknown severity.

ALRRSC Parameter
Specifies the resource name of alerts that are reporting problems, up to 50 resource names can be specified.

***ALL:** Shows alerts about all failing resources.

resource-name: Shows alerts that are reporting problems associated with the assigned resource name.

ALRRSCTYPE Parameter
Specifies the resource type(s) of alerts that are reporting problems, up to 50 resource types can be specified. Each resource name has a resource type associated with that resource. Resource types are diskette (DKT) or tape (TAP), for example.

***ALL:** Shows alerts for all resource types.

resource-type: Shows alerts that are reporting problems associated with the assigned resource type.

Example

```
WRKALR  DSPORT(*LOCAL)  ALRTYPE(*TEMP *PERM)  ALRRSCTYPE(DKT)
```

This command shows all locally created alerts in the alert file that are both temporary and permanent. The alerts shown are reporting problems about diskettes.

WRKAUTL (Work with Authorization Lists) Command

The Work with Authorization Lists (WRKAUTL) command shows a list of the authorization lists you want to work with. From this list you can display, edit, delete, display the list's objects, or change the text for an authorization list.

You must have ownership of the list to add, change, or remove users. You must own the authorization list to delete it.

```
                                                                  Required

                  ┌*ALL──────────────────────────┐ ┌─┐
                  │                              │ │P│
WRKAUTL───AUTL────┼authorization-list-name───────┼─┴─┴──────────────────────
                  └generic*-authorization-list-name┘

                                                   ┌──────────────────────┐
                                                   │Job: I  Pgm: I  Exec  │
                                                   └──────────────────────┘
```

AUTL Parameter
Specifies the name of an authorization list you want to work with.

***ALL:** A list of all the authorization lists that you own or have authority to see is shown.

authorization-list-name: Specify the authorization list that you want to work with.

generic-authorization-list-name:* Specify the generic name of the authorization list that you want to work with. A generic name can be specified as a character string that contains one or more characters followed by an asterisk (*), such as FR*. If a generic authorization list name is specified, then all authorization lists that have names with the same prefix as the generic authorization list name and that the user has some authority for are displayed.

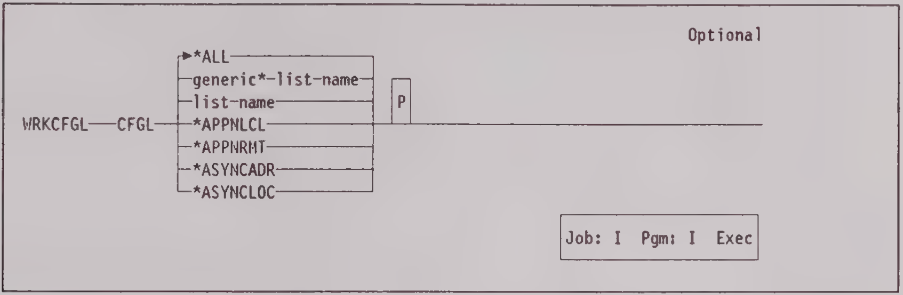
Example

```
WRKAUTL AUTL(FR*)
```

This command shows a list of all the authorization lists starting with FR that you have authority to see.

WRKCFGL (Work with Configuration Lists) Command

The Work with Configuration Lists (WRKCFGL) command provides an interactive interface to configuration list functions through the Work with Configuration Lists display. This command displays the Work with Configuration Lists display.



CFGL Parameter

Specifies the configuration list(s) to work with. You specify which configuration list(s) to include in the list of configuration lists on the Work with Configuration Lists display.

<u>*ALL:</u> You work with all configuration lists.

generic-list-name:* Specify configuration lists whose names start with a certain character string (generic name).

list-name: Specify a specific configuration list.

***APPNLCL:** You work with the advanced peer-to-peer networking (APPN) local location configuration list.

***APPNRMT:** You work with the APPN remote location configuration list.

***ASYNCADR:** You work with asynchronous packet assembly/disassembly (PAD) network address configuration lists.

***ASYNCLOC:** You work with the asynchronous remote location configuration list.

Example

```
WRKCFGL  CFGL(PEG*)
```

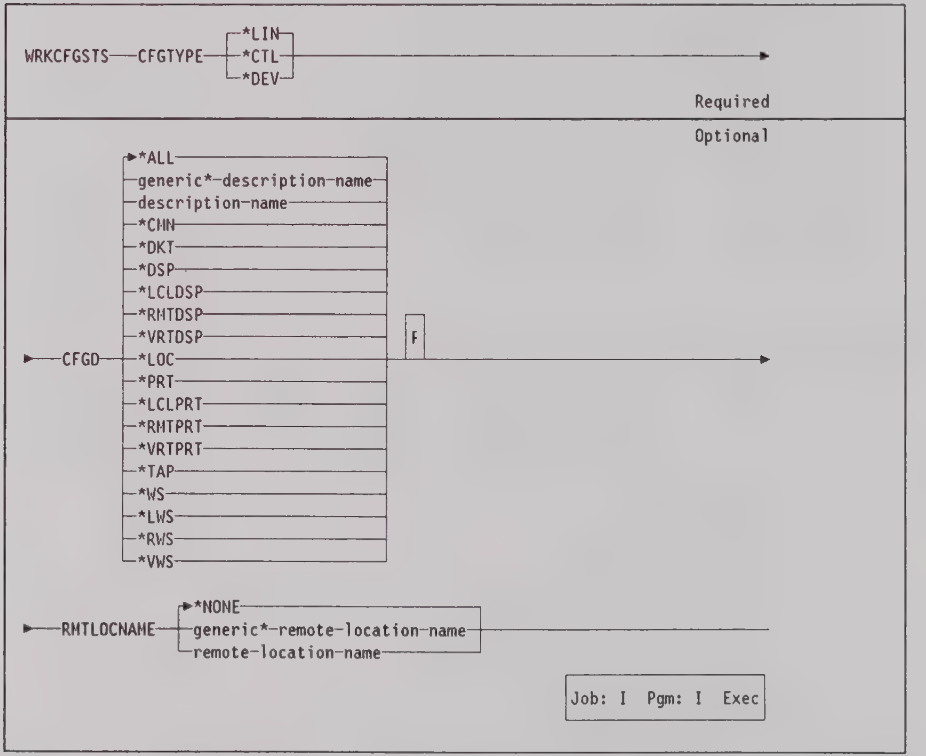
This command shows the Work with Configuration Lists display with entries for all configuration lists whose names start with PEG.

WRKCFGSTS (Work with Configuration Status) Command

The Work with Configuration Status (WRKCFGSTS) command provides an interactive interface to configuration status functions.

When this command is entered, the Work with Configuration Status display is shown. This display shows status information for lines, controllers, and/or devices, and for jobs associated with devices. The display can be for a location or for one or more lines, controllers, or devices. All attached configuration descriptions are displayed for each line, controller, or device description selected. The attached configuration descriptions are indented under the object they are attached to. If status is requested for a specific line, controller, or device, upline and downline attachments are shown. If status is requested for any collection of controllers or devices, only downline attachments are shown. Status for a location will display devices and modes for the specified location.

Options available on the Work with Configuration Status display are to vary status, end or resume recovery, hold or release a device, work with a job, and display location path. Options can be entered for more than one configuration object at a time. The option requests are processed sequentially in the order in which they appear in the list.



CFGTYPE
Specifies the type of description to display status for.

***LIN:** Display status for lines.

***CTL:** Display status for controllers.

***DEV:** Display status for devices.

CFGD Parameter
Specifies the descriptions you want displayed on the Work with Configuration Status display.

***ALL:** All descriptions of the type specified for the CFGTYPE parameter are displayed.

generic-description-name:* Specifies that all descriptions whose names begin with the letter(s) specified, and are of the type you specified for the CFGTYPE parameter, are displayed.

description-name: A specific description and any attachments, both upline as well as downline, are displayed.

***CMN:** Communications controllers and/or devices are displayed.

***DKT:** Diskette devices are displayed.

***DSP:** Both local and remote display station devices are displayed.

***LCLDSP:** Local display station devices are displayed.

***RMTDSP:** Remote display station devices are displayed.

***VRTDSP:** Virtual (passthru) display station devices are displayed.

***LOC:** Communications devices whose remote location name matches the name specified for the RMTLOCNAME parameter are displayed. A value, other than *NONE, must be specified for the RMTLOCNAME parameter if you specify *LOC for the CFGD parameter.

***PRT:** Both local and remote printer devices are displayed.

***LCLPRT:** Local printer devices are displayed.

***RMTPRT:** Remote printer devices are displayed.

***VRTPRT:** Virtual (passthru) printer devices are displayed.

***TAP:** Tape devices are displayed.

***WS:** Work station controllers are displayed.

***LWS:** Local work station controllers are displayed.

***RWS:** Remote work station controllers are displayed.

***VWS:** Virtual (passthru) work station controllers are displayed.

RMTLOCNAME Parameter

Specifies the remote location name of those communications devices for which status are displayed. Only those devices with the specified remote location name are listed on the Work with Configuration Status display.

This parameter is required if CFGD(*LOC) is specified. It is not a valid parameter for any other value of CFGD.

***NONE:** The descriptions displayed are not for communications devices with a specific location name. *NONE must be specified if the CFGD parameter is anything other than *LOC.

generic-remote-location-name:* Specifies the generic location name (starting characters of the name) of the communications device for which status is displayed.

remote-location-name: Specifies the remote location name of the communications devices for which status is displayed.

Example

```
WRKCFGSTS   CFGTYPE(*DEV)   CFGD(*RMTDSP)
```

This command shows the Work with Configuration Status display with the status for all remote display stations.

WRKCMD (Work with Commands) Command

The Work with Commands (WRKCMD) command shows a list of commands from a user specified subset of command names. From this list, several command-related functions will be available.

Restriction: The user must have authority to both the command and the library where it is located for the operation he wants to perform.

```
                                                                    Required
                 ┌►*LIBL/─────────┐
                 ├─*USRLIBL/──────┤  ┌─*ALL─────────────────┐  ┌─┐
WRKCMD───CMD─────┼─*ALLUSR/───────┼──┼─generic*-command-name─┼──┤P├──────────────
                 ├─*ALL/──────────┤  └─command-name─────────┘  └─┘
                 └─library-name/──┘

                                                     ┌─────────────────────┐
                                                     │Job: I  Pgm: I  Exec │
                                                     └─────────────────────┘
```

CMD Parameter

Specifies the qualified name of the command to be shown on the Work with Commands display. A specific command or a generic command can be specified; either type can be optionally qualified by a library name.

The possible library values are:

<u>*LIBL</u>: All libraries in the user and system portions of the job's library list are searched.

***USRLIBL:** Only the libraries listed in the user portion of the library list are searched.

***ALLUSR:** All nonsystem libraries, including all user-defined libraries and the QGPL library, not just those in the job's library list, are searched. Libraries whose names begin with the letter Q, other than QGPL, are not included.

***ALL:** All libraries in the system, including QSYS, are searched.

library-name: Only the library named in this parameter is searched.

***ALL:** All commands (in the libraries specified) are listed on the Work with Commands display.

generic-command-name:* Specify the generic name of the command being listed. A generic name can be specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified for the CMD parameter, then all commands that have names with the same prefix as the generic command name are listed.

For more information on the use of generic functions, refer to Appendix G, "Generic Functions."

command-name: Specify the name of the command being listed.

Example

```
WRKCMD CMD(QGPL/DSP*)
```

This command shows a list of all commands in library QGPL that start with DSP.

WRKCNTINF (Work with Contact Information) Command

The Work with Contact Information (WRKCNTINF) command is used to work with the information that helps you contact, or be contacted by, various support centers. When you specify this command, a display appears allowing you to select one of the four kinds of support functions.

More information on this command is in the *Q & A Data Base Coordinator's Guide.*

Restriction: This command is valid only in an interactive environment.

```
                                                            Required
WRKCNTINF───────────────────────────────────────────────────────────

                                                  Job: I  Pgm: I  Exec
```

There are no parameters for this command.

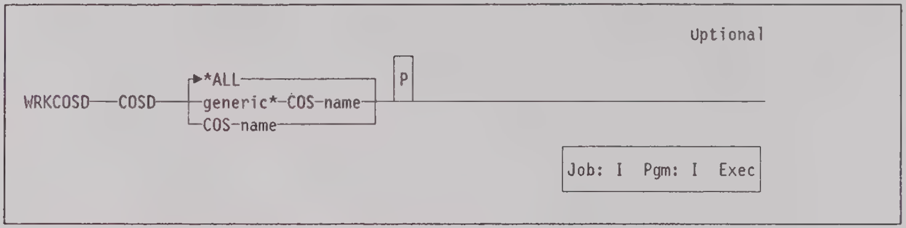
Example

```
WRKCNTINF
```

This command shows the Work with Support Contact Information display.

WRKCOSD (Work with Class-of-Service Descriptions) Command

The Work with Class-of-Service Descriptions (WRKCOSD) command provides an interactive interface to class-of-service description functions through the Work with Class-of-Service Descriptions display. This command displays the Work with Class-of-Service Description display.



COSD Parameter

Specifies the class-of-service description(s) to work with. Specify which class-of-service description(s) to include in the list of class-of-service descriptions on the Work with Class-of-Service Descriptions display.

*ALL: You work with all class-of-service descriptions.

generic-COS-name:* Specify that you will work with all class-of-service descriptions whose names start with a certain character string (generic name).

COS-name: Specify a specific class-of-service description.

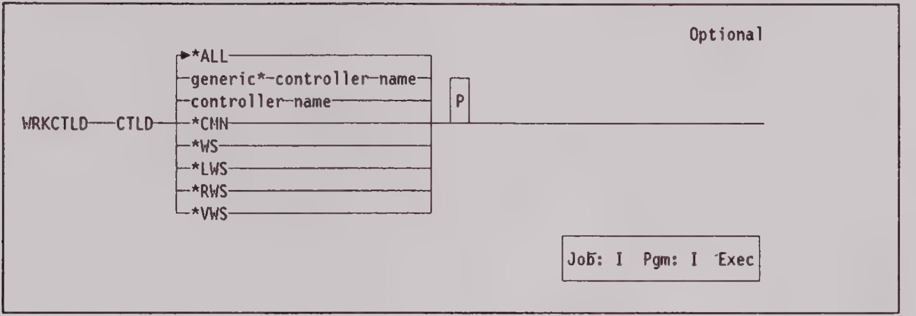
Example

```
WRKCOSD  COSD(MPLS*)
```

This command shows the Work with Class-of-Service Descriptions display with entries for all class-of-service descriptions whose names start with MPLS.

WRKCTLD (Work with Controller Descriptions) Command

The Work with Controller Descriptions (WRKCTLD) command provides an interactive interface to controller description functions through the Work with Controller Descriptions display. This command shows the Work with Controller Descriptions display.



CTLD Parameter
Specify which controller description(s) to include in the list of controller descriptions on the Work with Controller Descriptions display.

***ALL:** You work with all controller descriptions.

generic-controller-name:* Specify that you will work with controller descriptions whose names start with a certain character string (generic name).

controller-name: Specify a specific controller description.

***CMN:** You work with communications controller descriptions only.

***WS:** This value means you work with work station controller descriptions only.

***LWS:** You work with local work station controller descriptions only.

***RWS:** You work with remote work station controller descriptions only.

***VWS:** You work with virtual work station controller descriptions only.

Example

```
WRKCTLD  CTLD(*LWS)
```

This command shows the Work with Controller Descriptions display with entries for all local work station controllers.

WRKDBFIDD (Work with DB Files using IDDU) Command

The Work with Data Base Files using IDDU (WRKDBFIDD) command shows the IDDU Work with Data Base Files display where you can select an option to create physical files or enter data into a file.

```
                                                                    Required
WRKDBFIDD───────────────────────────────────────────────────────────────────
                                                       ┌────────────────────────┐
                                                       │Job: I   Pgm: I   Exec  │
                                                       └────────────────────────┘
```

There are no parameters for this command.

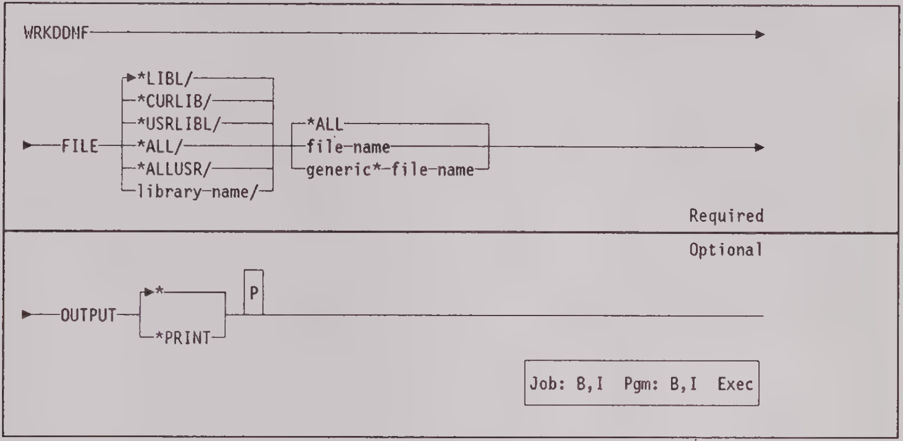
Example

```
WRKDBFIDD
```

This command shows the IDDU Work with Data Base Files display.

WRKDDMF (Work with Distributed Data Management File) Command

The Work with Distributed Data Management File (WRKDDMF) command displays a list of DDM files. From this list, you can change, delete, or create DDM files, you can also, display or print details or print a list of the DDM files.



FILE Parameter

Specifies the qualified name of the DDM file whose description is being changed.

Depending on the library qualifier specified or assumed, the following libraries (for which the user has the authority) are searched for the specified file:

- ***LIBL**: All the libraries in the user *and* system portions of the job's library list are searched. If a specific file name is given, only the first file found by that name is shown.
- ***CURLIB**: The current library for the job is searched. If no current entry exists in the library list, QGPL is used.
- ***USRLIBL**: Only the libraries listed in the user portion of the job's library list are searched. If a specific file name is given, only the first file found by that name is shown.
- ***ALL**: All the libraries in the system, including QSYS, are searched. All files that match the specified file name and that are found within one or more of the identified libraries are shown.
- ***ALLUSR**: All the nonsystem libraries which include *all* user-defined libraries and the QGPL library, not just those in the job's library list, are searched. Libraries that begin with the letter Q, other than QGPL, are not included. All files that match the specified file name and that are found within one or more of the identified libraries are shown.
- *library-name*: Only the library named in this parameter is searched. The user must have read authority for the specified library.

***ALL:** All files in the library or libraries specified by the library qualifier are to have their descriptions shown.

file-name: Specify the qualified full name of the file that has its description shown.

generic-file-name:* Specify the qualified generic name of the file or files that are to have their descriptions shown. To specify a generic file name, add an asterisk (*) at the end of the character string common to the names of all the files desired. For more information on changes made to files with generic file names, refer to Appendix G, "Generic Functions."

OUTPUT Parameter

Specifies if the output from the command is shown at the requesting work station or listed with the job's spooled output on a printer. (Refer to Appendix A, "Expanded Parameter Descriptions" for an expanded description of the OUTPUT parameter, and to Appendix D, "Files Used by CL Commands" for the names of the files used by this command.)

***:** The output is shown (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

***PRINT:** The output is listed with the job's spooled output on a printer.

Example

```
WRKDDMF
```

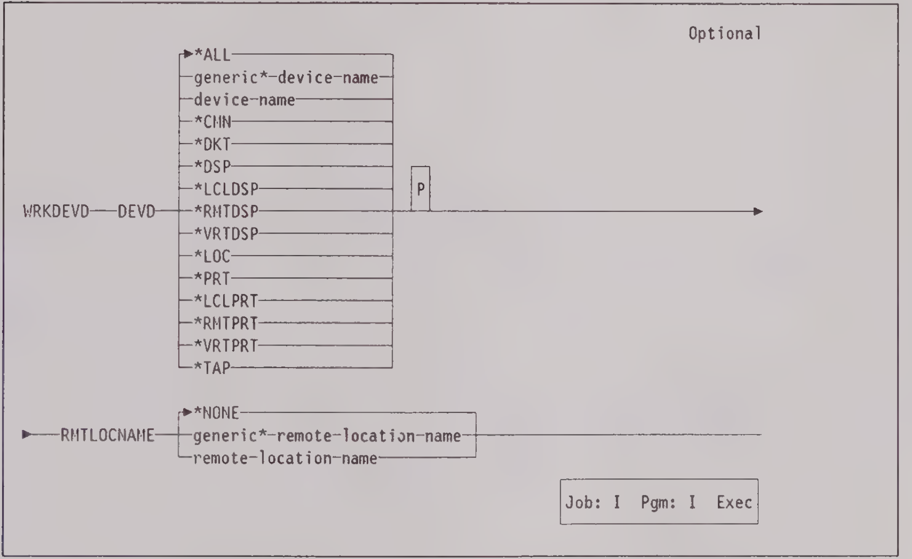
This command shows the Work with DDM Files display.

WRKDEVD (Work with Device Descriptions) Command

The Work with Device Descriptions (WRKDEVD) command provides an interactive interface to device description functions. When this command is entered, the Work with Device Descriptions display is shown. This display consists of a list of device descriptions.

From the Work with Device Descriptions display, you can perform all of the available configuration functions with your device descriptions. You can create, change, copy, delete, display, and print your device descriptions.

This command also allows you to group the device descriptions that appear on the Work with Device Descriptions display. You can have all of the device descriptions appear on the display, or you can have only those device descriptions of a certain type appear on the display, or only those device descriptions whose names start with a certain character string appear on the display.



DEVD Parameter
Specifies the device description(s) you want to have listed on the Work with Device Descriptions display.

***ALL:** All device descriptions are displayed.

generic-device-name:* All device descriptions whose names begin with the letter(s) specified are displayed.

device-name: Only the specified device description is displayed.

***CMN:** Communications devices are displayed.

***DKT:** Diskette devices are displayed.

***DSP:** Both local and remote display station devices are displayed.

***LCLDSP:** Local display station devices are displayed.

***RMTDSP:** Remote display station devices are displayed.

***VRTDSP:** Virtual (passthru) display station devices are displayed.

***LOC:** Devices whose remote location name matches the name specified on the RMTLOCNAME parameter will be displayed. A value, other than *NONE, must be specified for the RMTLOCNAME parameter if you specify *LOC for the DEVD parameter.

***PRT:** Both local and remote printer devices are displayed.

***LCLPRT:** Local printer devices are displayed.

***RMTPRT:** Remote printer devices are displayed.

***VRTPRT:** Virtual (passthru) printer devices are displayed.

***TAP:** Tape devices are displayed.

RMTLOCNAME Parameter
Specifies the remote name for those communications devices to be displayed. Only those device descriptions with the specified remote location name

are listed on the Work with Device Descriptions display.

This parameter is required if DEVD(*LOC) is specified. It is not a valid parameter for any other value of DEVD.

<u>*NONE:</u> The descriptions displayed are not for any specific location. *NONE must be specified if the DEVD parameter is anything other than *LOC.

generic-remote-location-name:* Specifies the generic location name (starting characters of the name) of the communications device which will be displayed.

remote-location-name: Specifies the remote location name of the communications device(s) which will be displayed.

Example

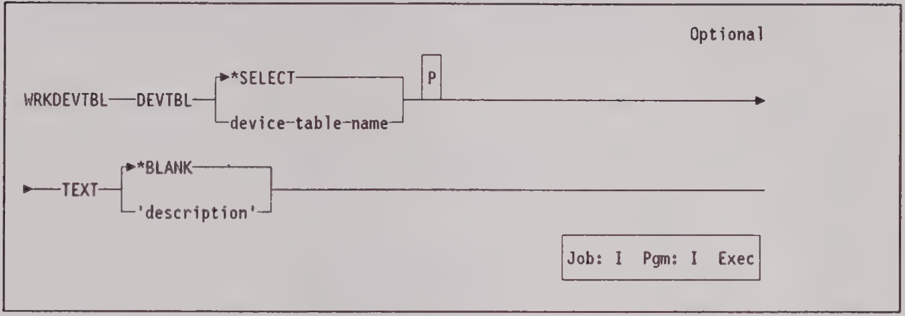
```
WRKDEVD   DEVD(*LCLPRT)
```

This command shows the Work with Device Descriptions display with all of the local printers listed.

WRKDEVTBL (Work with Device Table) Command

The Work with Device Table (WRKDEVTBL) command lets you create finance device tables and, once they are created, lets you add or delete device names in these tables. Several finance device tables can be defined, but you must use a unique name for each table. If the device table you specify on this command does not exist, it is created.

Device table updates can be accessed by any finance job that is submitted after all changes are completed.



DEVTBL Parameter
Specifies the name of a device table that contains 4704 or 3624 device names.

***SELECT:** Displays the list of existing device tables. Through this display you can add new device tables to the system, delete old device tables from the system, or change the contents of existing device tables.

device-table-name: Specify the name of the device table that you want to work with.

TEXT Parameter
Specifies the user-defined text that describes the device table. (For an expanded description of the TEXT parameter, see Appendix A, “Expanded Parameter Descriptions.”)

***BLANK:** No text is to be specified.

'description': Specify no more than 50 characters, enclosed in apostrophes. If you enter text in this field and specify DEVTBL(*SELECT), the text you enter is carried over to the Work with Device Tables display.

Examples

```
WRKDEVTBL DEVTBL(*SELECT)
```

This command shows the names of all finance device tables. You can add a new table, select an existing table for update, or remove tables on the display.

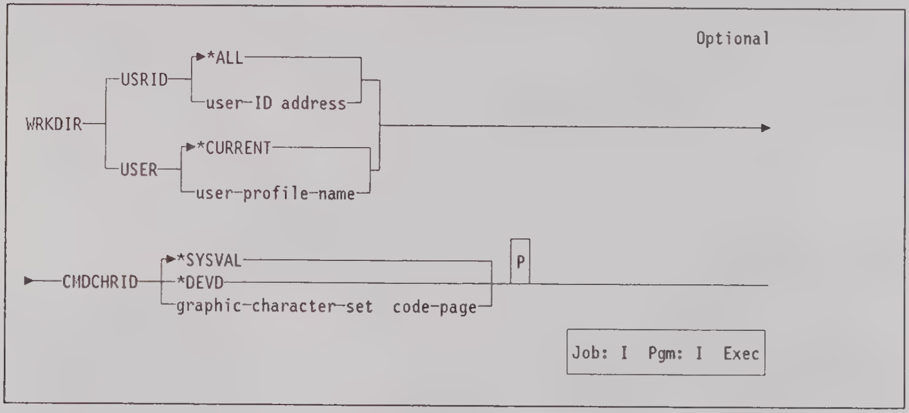
```
WRKDEVTBL  DEVTBL(DEVTBL1)  TEXT('Device group one')
```

This command shows the device table DEVTBL1, allowing you to add, change, or remove table entries. If DEVTBL1 does not exist when the WRKDEVTBL command is entered, a new device table is created using the specified text.

WRKDIR (Work with Directory) Command

The Work with Directory (WRKDIR) command provides a set of displays that allow a user to view, add, change, and remove entries in the distribution directory. When the WRKDIR command is entered, the system shows one or all of the entries in the system distribution directory, depending on the parameters specified. If the parameter specified applies to more than one directory entry, the system displays a list of directory entries. If the parameter identifies a specific directory user, the system displays a list of entries for that user.

Restriction: Users of this command must have security administrator authority to update all entries in the directory. Restrictions to what data can be updated apply when a nonadministrator runs this command. General access to view and print the directory is provided by the DSPDIR command.



USRID Parameter

Specifies which directory entries to display, printed, or output to a data base file by user ID and address. If USRID is specified, USER cannot be specified. More information on specifying user ID and address is in the *Communications: Distribution Services Network Administrator's Guide.*

***ALL:** Displays all directory entries in the system distribution directory. The entries are displayed in alphabetic order by user ID and address.

user-ID address: Specify the user ID and address of the directory entry to display.

USER Parameter

Specifies which directory entries to display associated with a particular user profile. If the user profile has no directory entries associated with it, an error message is sent. If USER is specified, USRID cannot be specified.

***CURRENT:** Specifies that the distribution directory entry of the user profile of the person currently running this command is displayed.

user-profile-name: Specify the user profile name of a directory entry displayed. A valid value is 1 to 8 characters and must be separated by at least one blank space.

CMDCHRID Parameter

Specifies the character identifier (graphic character set and code page) for the data being entered as command parameter values (applies to the USRID parameter). The character identifier is related to the display device that was used to enter the command. More information about CHRID processing is in the *Data Management Guide.*

***SYSVAL:** The system determines the graphic character set and code page values for the command parameters from the QCHRID system value.

***DEVD:** The system determines the graphic character set and code page values for the command parameters from the display device description where this command was entered. This option is supported only when the command is entered from an interactive job.

graphic-character-set code-page: Specify the graphic character set and code page values that were used to create the command parameters. Both values contain up to 3 characters and must be separated by at least one space.

Examples

```
WRKDIR  USRID(HURST PAYROLL)
```

Assume the user who is running this command has administrator authority. If the user ID and address of HURST PAYROLL exists in the directory, the Work with Directory Entries display is shown listing all entries for HURST PAYROLL.

```
WRKDIR  USER(JONES)
```

Assume the user who is running this command has security administrator authority. If the user profile of JONES exists in the directory, the Work with Directory Entries display is shown listing all entries with user profile of JONES.

```
WRKDIR
```

Assume the user who is running this command has security administrator authority. The Work with Directory Entries display is shown listing all entries in the directory.

```
WRKDIR
```

Assume the user who is running this command does not have security administrator authority. The Change Your Directory Details display is shown for this user. A message appears on the message line of this display indicating that this user is authorized only to change his own directory entry.

Additional Considerations

If the USRID parameter specifies the user ID and address of a user not currently contained in the distribution directory, an error message is returned. Likewise, if the USER parameter specifies a user profile not contained in the directory, an error message is returned.

This command allows a nonadministrator to update certain fields for the directory entry of that user only. If a nonadministrator runs this command and specifies a USRID or USER parameter other than himself, an error message is included on the Change Your Directory Details display indicating that the user is not authorized to update entries other than his own.

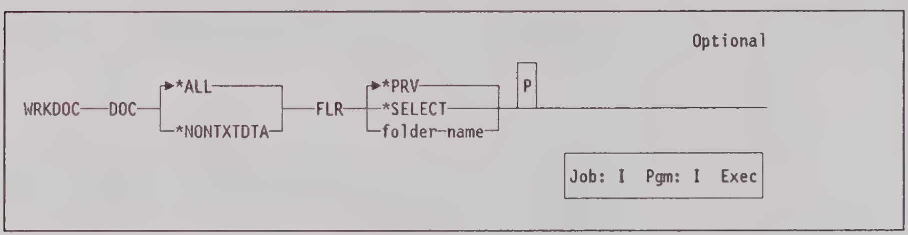
WRKDOC (Work with Documents) Command

The Work with Documents (WRKDOC) command calls the word processing function of the office product to show the Work with Documents display or the Work with Nontext Document Data display.

From the Work with Documents display, the user can select an option to: create, revise, copy, delete, view, print, rename, describe, print with options, send, spell check, file, paginate, and put security on documents.

From the Work with Nontext Document Data display, the user can select an option to: copy, delete, and rename nontext data, such as charts and images.

More information on working with documents is in the *Word Processing User's Guide*.



DOC Parameter
Specifies which display is shown.

***ALL:** The Work with Documents display is shown.

***NONTXTDTA:** The Work with Nontext Document Data display is shown.

FLR Parameter
Specifies the name of the folder used on the Work with Documents display or Work with Nontext Document Data display.

***PRV:** The folder named in your last session is used.

***SELECT:** Displays a list of folders from which you can select a folder.

folder-name: Specify the name of the folder you want to work with on the specified display.

Example

```
WRKDOC DOC(*ALL) FLR(*SELECT)
```

This command shows the Work with Documents display and a list of folders from which you can select the folder you want to work with.

WRKDOCLIB (Work with Document Libraries) Command

The Work with Document Libraries (WRKDOCLIB) allows the user to manage the Document Interchange Architecture (DIA) Library services available on remote systems in the network. Each remote document library that services are obtained for can be configured. This includes indicating the level of DIA services the remote system gives and associating the remote document library with a SNADS queue on which work requests are place.

Restriction: You must have security officer or security administrator authority to use this command.

```
WRKDOCLIB────────────────────────────────────────────────────────
                                                ┌─────────────────────┐
                                                │Job: I  Pgm: I  Exec │
                                                └─────────────────────┘
```

There are no parameters for this command.

Example

```
WRKDOCLIB
```

This command shows the Work with Document Libraries display.

WRKDOCPRTQ (Work with Document Print Queue) Command

The Work with Documents on the Print Queue (WRKDOCPRTQ) command calls the word processing function of the office product to show the Work with Documents to Be Printed display. From this display, users can manage their printed output.

More information on word processing is in the *Word Processing User's Guide*.

```
                                                                  Optional
WRKDOCPRTQ──────────────────────────────────────────────────────────────

                                                       Job: I  Pgm: I  Exec
```

There are no parameters for this command.

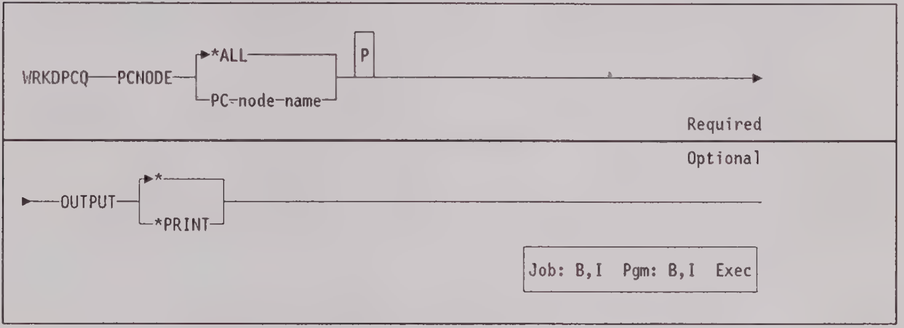
Example

```
WRKDOCPRTQ
```

This command shows the Work with Documents to Be Printed display.

WRKDPCQ (Work with DSNX/PC Distribution Queues) Command

The Work with DSNX/PC Distribution Queues (WRKDPCQ) command performs some simple operations on the DSNX/PC distribution queues. These queues are where DSNX distribution that are bound for a PC (locally attached to the AS/400 system and configured in the system directory as a DSNX-PC node) are held. A PC running DSNX-PC initiates the DS-SEND function that sends the queued distributions to that requesting PC.



PCNODE Parameter
Specifies the PC node names that queue entries are displayed for.

***ALL:** Specifies that all the PC nodes that currently have queued entries are displayed.

PC-node-name: The specified node and its entries are displayed.

OUTPUT Parameter
Specifies if the output from the command is shown at the requesting work station or printed.

***:** The output is shown (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

***PRINT:** The output is listed with the job's spooled output on a printer.

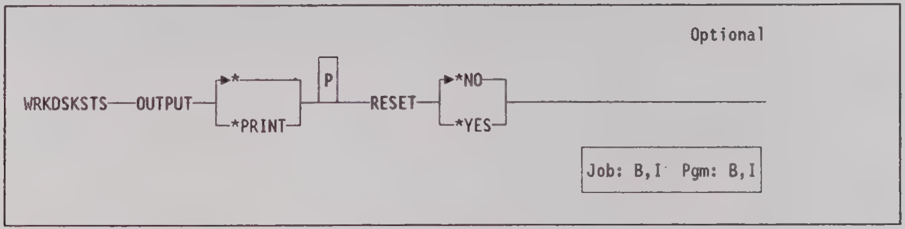
Example

```
WRKDPCQ
```

This command shows the Work with DSNX/PC Queues display.

WRKDSKSTS (Work with Disk Status) Command

The Work with Disk Status (WRKDSKSTS) command displays performance and status information for the disk units on the system.



OUTPUT Parameter

Specifies whether the output from the command is displayed at the requesting work station or listed with the job's spooled output on a printer.

***:** The output is displayed (if requested by an interactive job) or listed with the job's spooled output (if requested by a batch job).

***PRINT:** The output is listed with the job's spooled output on a printer.

RESET Parameter

Specifies whether the disk statistics are reset.

***NO:** The disk statistics are not reset. The measurement time interval is extended if a previous display disk status command has started in the current job.

***YES:** The disk statistics are reset. A measurement time interval of zero is used.

Example

```
WRKDSKSTS  OUTPUT(*PRINT)
```

This command directs the disk information to the job's spooled output on a printer. If OUTPUT(*) is specified instead and the command was entered from a work station, the information about the disk units is displayed at the work station.

WRKDSTL (Work with Distribution List) Command

The Work with Distribution List (WRKDSTL) command provides a set of displays that allows a user to view, add, remove, and change distribution lists. A distribution list is a list of directory entries used to simplify the sending of distributions to a group of users. When the WRKDSTL command is entered, the system shows one, some, or all of the distribution lists in the system directory, depending on the value specified on the LSTID parameter. (LSTID(*ALL) specifies that all distribution lists be displayed; LSTID followed by a specific list ID specifies that a specific distribution list be displayed.)

Restriction: Users of this command must have security administrator authority to change another user's distribution list. (Users can change distribution lists that they create without any special authorities; any user can create a distribution list.)

```
                                                                Optional
                                                    ┌─┐
                    ┌►*ALL──────────────────┐       │P│
WRKDSTL───LSTID─────┤                       ├───────┴─┴──────────────────────────►
                    └─list-identification───┘

                 ┌►*SYSVAL─────────────────────────────┐
►──CMDCHRID──────┼─*DEVD───────────────────────────────┼──────────────────────────
                 └─graphic-character-set   code-page───┘

                                                          ┌─────────────────────┐
                                                          │Job: I  Pgm: I  Exec │
                                                          └─────────────────────┘
```

LSTID Parameter

Specifies which distribution lists to display.

***ALL:** Displays all distribution lists in the system distribution directory. The entries are displayed in alphabetical order by list ID.

list-identification: Specify the two-part name of an existing distribution list. Each part must be 1 to 8 characters in length and the two parts must be separated by at least one space.

Note: The naming rules for the two-part list (ID) are identical to the rules for the user ID and address. A complete description of these rules is in the *Communications: Distribution Services Network Administrator's Guide.*

CMDCHRID Parameter

Specify the character identifier (graphic character set and code page) for the data being entered as command parameter values (applies to the LSTID parameter). The character identifier is related to the display device that was used to enter the command. More information about CHRID processing is in the *Data Management Guide.*

***SYSVAL:** The system determines the graphic character set and code page values for the command parameters from the QCHRID system value.

***DEVD:** The system determines the graphic character set and code page values for the command parameters from the display device description where this command was entered. This option is supported only when the command is entered from an interactive job. If specified in a batch job, an error occurs.

graphic-character-set code-page: Specify the graphic character set and code page values that were used to create the command parameters. Both values contain up to 3 characters and must be separated by at least one space.

Examples

```
WRKDSTL LSTID(WILL DISTLIST)
```

The Work with Distribution Lists display is shown with one entry of WILL DISTLIST.

```
WRKDSTL
```

The Work with Distribution Lists display shows a list of all distribution lists in the distribution directory.

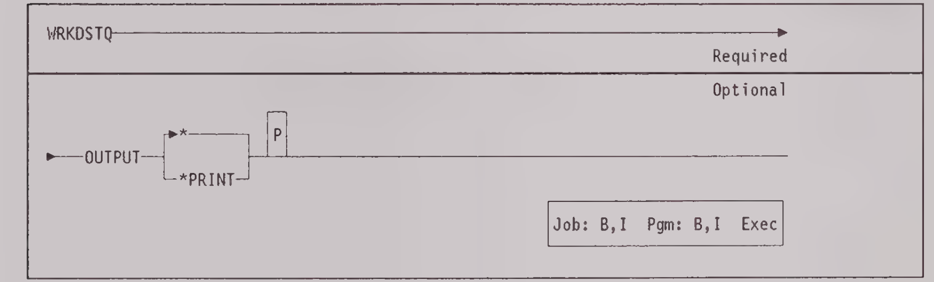
Additional Considerations

If the LSTID parameter specifies a list (ID) not currently contained in the distribution directory, an error message is returned.

WRKDSTQ (Work with Distribution Queue) Command

The Work with Distribution Queue (WRKDSTQ) command displays and controls the distribution requests on the systems network architecture service (SNADS) next system queues. A detailed description of the SNADS network is in the *Communications Programmer's Guide*.

Restrictions: (1) This command cannot be used on an IBM 5251 Model 1 or 2. (2) Before this command is run for the first time, the QSNADS subsystem must be started to create the internal SNADS objects that this command uses. (3) Messages that report errors about node identifiers may display or print different characters than the user entered because of internal system transformations. The internal value for a node identifier may differ from the characters shown by the DSPDSTSTS command depending on the type of work station (language) you are using. If the byte-string value specified for the NDEID parameter does not match the rules for an internal node identifier value or if it does not match the internal value for any defined node (ignoring case differences), an error may be reported.



OUTPUT Parameter
Specifies if the output from the command is displayed at the requesting work station or listed with the job's spooled output on a printer. (Refer to Appendix A, "Expanded Parameter Descriptions" for an expanded description of the OUTPUT parameter.)

***:** The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

***PRINT:** The output is listed with the job's spooled output on a printer.

Example

```
WRKDSTQ
```

This command displays the status of all system queues displayed.

WRKDTADCT (Work with Data Dictionaries) Command

The Work with Data Dictionaries (WRKDTADCT) command shows the IDDU Work with Data Dictionaries display where you can select an option to create, change, delete, or print the contents of a data dictionary.

```
                                                         Required
WRKDTADCT──────────────────────────────────────────────────────────
                                                ┌──────────────────────┐
                                                │Job: I  Pgm: I  Exec  │
                                                └──────────────────────┘
```

There are no parameters for this command.

Example

```
WRKDTADCT
```

This command shows the Work with Data Dictionaries display.

WRKDTADFN (Work with Data Definitions) Command

The Work with Data Definitions (WRKDTADFN) command displays the IDDU Select Definition Type display. From this display you select which dictionary and which type of definition to process. After making these selections, you can create, copy, delete, print, or rename definitions or display where definitions are used.

```
                                                     Required
WRKDTADFN----------------------------------------------------

                                           Job: I  Pgm: I  Exec
```

There are no parameters for this command.

Example

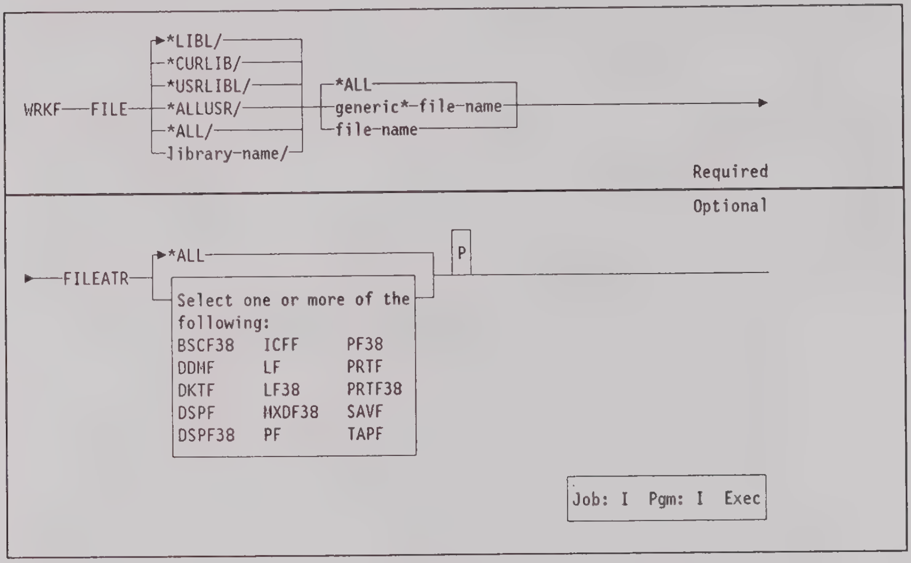
```
WRKDTADFN
```

This command shows the Select Definition Type display.

WRKF (Work with File) Command

The Work with File (WRKF) command shows a list of files you have authority to use. From this list you can copy, delete, display descriptions, save, and restore files.

Restriction: Before you can display the files you have specified, you must have object operational authority to those files. Also, of the libraries identified by the library qualifier, only those libraries that you have *USE authority to are searched for the specified files.



FILE Parameter

Specifies which files in the library or libraries are shown. If no library qualifier is specified, *LIBL is assumed and all libraries in the job's library list are searched for the files.

Depending on the library qualifier specified or assumed, the following libraries (for which the user has the authority) are searched for the specified files:

- **_*LIBL_**: All the libraries in the user and system portions of the job's library list is searched.
- ***CURLIB**: The current library for the job is searched. If no current entry exists in the library list, QGPL is used.
- ***USRLIBL**: Only the libraries listed in the user portion of the job's library list is searched.
- ***ALLUSR**: All the nonsystem libraries, which include all user-defined libraries and the QGPL library, that are not in the job's library list is searched. Libraries whose names start with the letter Q, other then QGPL, are not included.
- *ALL: All of the libraries in the system, including QSYS, is searched.
- *library-name*: Only the library named is searched. The user must have *USE rights for the specified library.

Only those files that the user has authority to is searched.

***ALL:** All files in the libraries identified in the library qualifier are shown. The user can display only those files for which he has some authority.

generic-file-name:* Specify the generic file name that is shown. To specify the generic name, add an asterisk (*) after the last character in the generic name (for example, ABC*). If an * is not included with the name, the system assumes that the name is a complete file name.

file-name: Specify the qualified name of the file that is shown. If the library qualifier is *ALL, *ALLUSER, or a library name, all files of the specified types that the user has some authority to (for example, operational rights) and that are in the specified libraries are shown.

FILEATR Parameter

Specifies the type of file whose attributes are shown.

*ALL: The attributes of all files are shown.

BSCF38: The attributes of the binary synchronous communication file for a System/38 are shown.

DDMF: The attributes of the distributed data management (DDM) file are shown.

DKTF: The attributes of diskette files are shown.

DSPF: The attributes of display files are shown.

DSPF38: The attributes of display files for a System/38 are shown.

ICFF: The attributes of communications files are shown.

LF: The attributes of logical files are shown.

LF38: The attributes of logical files for a System/38 are shown.

MXDF38: The attributes of mixed files for a System/38 are shown.

PF: The attributes of physical files are shown.

PF38: The attributes of physical files for a System/38 are shown.

PRTF: The attributes of printer files are shown.

PRTF38: The attributes of printer files for a System/38 are shown.

SAVF: The attributes of save files are shown.

TAPF: The attributes of tape files are shown.

Example

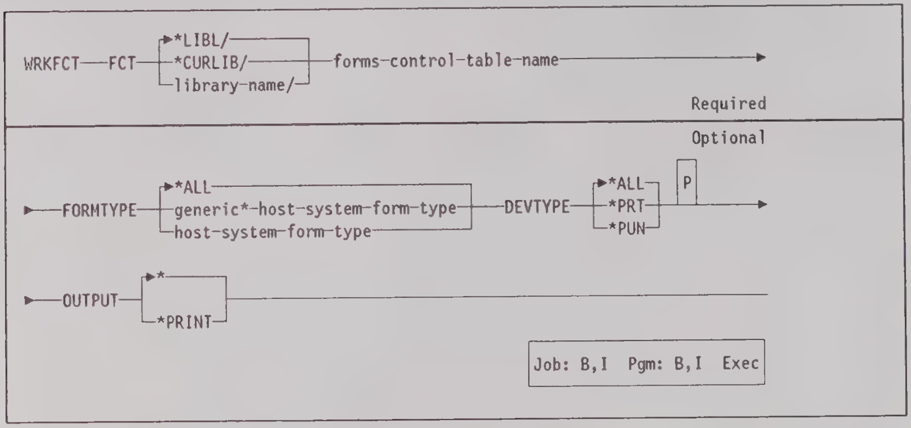
```
WRKF   FILE(X/PAY)
```

The file (for which the user has some authority) that is named PAY and is located in the library X is displayed.

WRKFCT (Work with Forms Control Table) Command

The Work with Forms Control Table (WRKFCT) command displays the forms control table (FCT) entries. You can show, change, or remove the entries. The output is shown on the display or printed.

Restriction: To use this command, you must have read authority to the FCT and the library where the FCT is stored.



FCT Parameter
Specifies the qualified name of the FCT to work with.

The possible library values are:

***LIBL:** The library list is used to locate the FCT.

***CURLIB:** The current library for the job is used to locate the FCT. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the FCT is located.

FORMTYPE Parameter
Specifies the host system form type associated with the FCT.

***ALL:** Specifies all form types.

generic-host-system-form-type:* Specify the generic host system form type. To specify a generic host system form type, add an asterisk after the last character in the generic name (ABC*, for example). If an * is not included with the name, the system assumes that the name is complete.

host-system-form-type: Specify the host system form type.

DEVTYPE Parameter
Specifies the device type associated with the FCT.

***ALL:** Specifies all device types.

***PRT:** Specify a printer device.

***PUN:** Specify a punch device.

OUTPUT Parameter
Specifies if the output from the command is shown at the requesting work station or printed with the spooled output of the job.

***:** Output is shown on the display if it is requested by an interactive job. Output is printed with the spooled output of the job if it is requested by a batch job.

***PRINT:** Output is printed with the spooled output of the job. The IBM-supplied device file QPWRKFCT is used to print the FCT.

Example

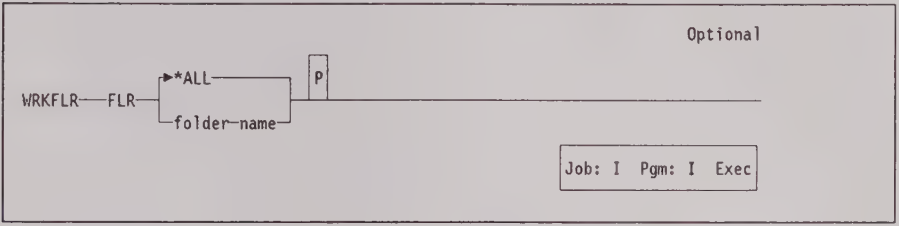
```
WRKFCT  FCT(USERLIB/FORMCTRL)  FORMTYPE(MED*)  +
  DEVTYPE(*PRT)  OUTPUT(*)
```

This command (if entered from a batch job) sends selected forms control entries to the spooling queue of the job to print. If the command is entered in an interactive job, a menu of selected forms control entries is displayed and selections can be made. The menu contains printer entries that begin with the characters MED.

WRKFLR (Work with Folders) Command

The Work with Folders (WRKFLR) command calls the word processing function of the office product to show the Work with Folders display. From this display, the user can select an option to: create, delete, rename, describe, and put security on a folder, and to work with documents.

More information on working with folders is in the *Word Processing User's Guide.*



FLR Parameter
Specifies the name of the folder used on the Work with Folders display. The list shown is always the list of folders the user has authority to.

***ALL:** The list of folders that consist of all of the root level or first level folders is displayed.

folder-name: Specify the name of the folder to display the folders contained in the requested folder.

Example

```
WRKFLR  FLR(*ALL)
```

This shows the Work with Folders display. A list of all folders you are authorized to use is shown.

WRKHDWPRD (Work with Hardware Products) Command

The Work with Hardware Products (WRKHDWPRD) command displays a menu panel with options to display the rack configuration, copy the rack configuration, or replace the rack configuration.

When option 1, Work with rack configuration, is chosen, WRKHDWPRD determines if any new I/O feature cards, disk units, diskette units, or tape units have been added to the system. If any of these have been added or moved, you are to enter the rack identifier and vertical location of the containing card enclosure. If a new rack or card enclosure is determined to exist, you are to enter the attributes of these items.

If there have been no additions of disk, diskette, or tape units or I/O feature cards, a display showing the rack hardware inventory (sorted by rack) is shown. You can do the following on this display:

- Change items (option 2 on item line)
- Delete items (option 4 on item line)
- Print the rack inventory list and a diagram of the rack and enclosed devices and features (F17)
- Call a display to add filler displays and non-IBM hardware items (F15)
- Call a display to work with communications lines (F14)
- Call a display to display cables (F18)

```
                                                              Required
WRKHDWPRD─────────────────────────────────────────────────────────────
                                                 ┌────────────────────┐
                                                 │Job: I  Pgm: I  Exec│
                                                 └────────────────────┘
```

There are no parameters for this command.

Example

```
WRKHDWPRD
```

This command shows the Work with Hardware Products display.

WRKJOB (Work with Job) Command

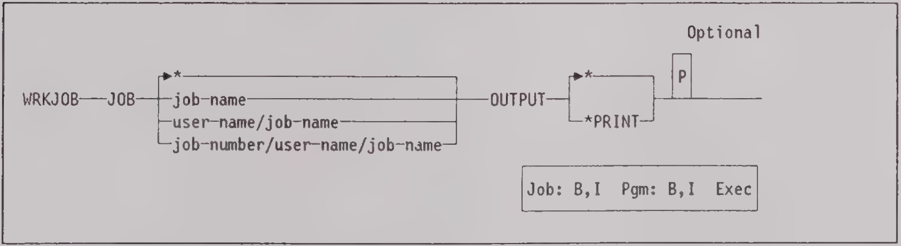
The Work with Job (WRKJOB) command allows you to work with or change information concerning a job.

A user can work with or change the following:

- Job status attributes
- Job definition attributes
- Job run attributes
- Spooled file information
- Job log information
- Program stack information
- Job lock information
- Library list information
- Open file information
- File override information
- Commitment control status

The information for the following options can be shown only when the job is active: job run attributes, program stack information, job lock information, library list information, job log information, open file information, file override information, and commitment control status. The following options: job status attributes, job definition attributes, and spooled file information can be found regardless of where the user's job is in the system: on the job queue, on an output queue, or active in the system. Note, however, that the job is not considered to be in the system until all of its input has been completely read in; only then is an entry placed on the job queue.

Restriction: You can work with only your own job unless you have the special job control rights (*JOBCTL) or unless the job you want to work with has the same user name you have.



JOB Parameter

Specifies the name of the job whose information is being worked with.

*: The job whose information is displayed is the job where this work with command is issued.

job-name: Specify the name of the job that has information about it displayed. If no job qualifier is given, all of the jobs currently in the system are searched for the simple job name. If duplicates of the specified name are found, a list of messages containing the qualified job names of all duplicates is displayed. (For an expanded description of the JOB parameter and nonunique job names, see Appendix A, "Expanded Parameter Descriptions.")

OUTPUT Parameter

Specifies if the display is shown that allows you to work with or change a job displayed at the requesting work station or printed with the job's spooled output. (Refer to Appendix A, "Expanded Parameter Descriptions" for an expanded description of the OUTPUT parameter, and to Appendix D, "Files Used by CL Commands" for the name of the printer file used by this command.)

*: The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

***PRINT:** The output is printed with the job's spooled output.

Example

```
WRKJOB  JOB(SMITH/PAYROLL)  OUTPUT(*PRINT)
```

This command directs the information for the job named PAYROLL submitted by the user named SMITH to the job's output spooling queue for printing.

WRKJOBD (Work with Job Description) Command

The Work with Job Description (WRKJOBD) command allows you work with a list of job descriptions. Job descriptions determine how you want the system to process jobs.

Restriction: You must have object operational and read authorities for the job description before you can display the list of job descriptions.

```
                                                                  Required
                  ┌►*LIBL/──────────────┐
                  ├─*CURLIB/────────────┤
                  ├─*USRLIBL/───────────┤
WRKJOBD───JOBD────┼─*ALLUSR/────────────┼──────────────────────────────────►
                  ├─*ALL/───────────────┤
                  └─library-name/───────┘

                                      ┌─┐
   ┌─*ALL──────────────────────────┐  │P│
►──┼─job-description-name──────────┼──┴─┴───────────────────────────────────
   └─generic*-job-description-name─┘

                                               ┌─────────────────────┐
                                               │Job: I  Pgm: I  Exec │
                                               └─────────────────────┘
```

JOBD Parameter

Specifies the qualified name of the job descriptions that is shown.

The possible library values are:

<u>*LIBL:</u> The library list is used to locate the job description.

***CURLIB:** The current library for the job is used to locate the job description. If no current library entry exists in the library list, QGPL is used.

***USRLIBL:** Only the libraries listed in the user portion of the job's library list are used to locate the JOBD.

***ALLUSR:** All the nonsystem libraries, which include all user-defined libraries and the QGPL library are used to locate the JOBD. Libraries whose names start with the letter Q, other then QGPL, are not included.

***ALL:** All of the libraries in the system, including QGPL, are searched.

library-name: Specify the library where the job description is located.

The possible JOBD values are:

***ALL:** Specifies that all job descriptions in the library specified are shown.

job-description-name: Specify the name of the job description shown.

generic-job-description-name:* Specify the generic name of the job description shown. For example, if you type T* in the library QGPL, you get a list of all the job descriptions starting with T in library QGPL.

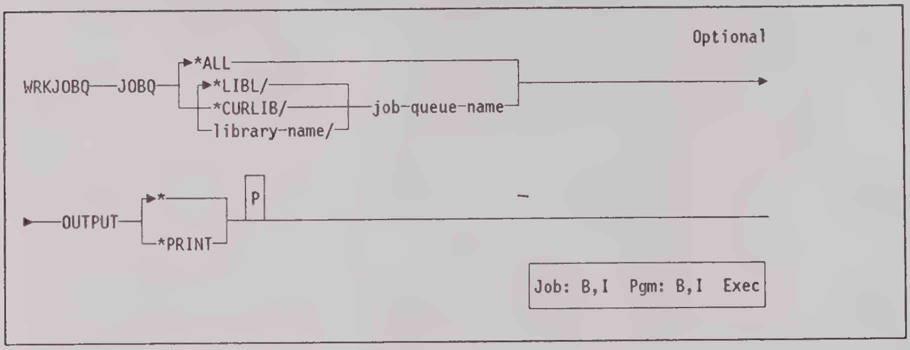
Example

```
WRKJOBD  JOBD(MYLIB/SPEC*)
```

This command shows a list of job description names beginning with SPEC in library MYLIB. Options may be selected from this display to work with the job descriptions that are listed.

WRKJOBQ (Work with Job Queue) Command

The Work with Job Queue (WRKJOBQ) command allows you to work with the overall status of all job queues or the detailed status of a specific job queue. The status of the queues may change while the command is being run.



JOBQ Parameter

Specifies that all job queues are shown, or specifies the qualified name of the job queue whose status is shown.

***ALL:** The overall status of all job queues is shown.

The possible library values are:

***LIBL:** The library list is used to locate the job queue.

***CURLIB:** The current library for the job is used to locate the job queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the qualified library where the job queue is located.

job-queue-name: Specify the qualified name of the job queue that detailed status information is shown for.

OUTPUT Parameter

Specifies whether the output from the command is shown at the requesting work station or printed with the job's spooled output on a printer. (Refer to Appendix A, "Expanded Parameter Descriptions" in Volume 1 for an expanded description of the OUTPUT parameter, and to Appendix D, "Files Used by CL Commands" in Volume 1 for the name of the printer file used by this command.)

***:** The output is shown (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

***PRINT:** The output is printed with the job's spooled output on a printer.

Example

```
WRKJOBQ  JOBQ(QGPL/QBATCH)
```

This command shows the detailed status information about the job queue named QBATCH in library, QGPL. Each job on the QBATCH job queue is identified by job name, user name, and job number; the job's priority and status are also shown.

WRKJRN (Work with Journal) Command

The Work with Journal (WRKJRN) command displays a menu from which you can select options for journal operations. From the primary menu, options can be selected to display the status of the journal, do various types of recovery, display a list of all journals on the system, or prompt for and enter any of the journal commands.

```
                                                          Required
WRKJRN─────────────────────────────────────────────────────────────

                                                ┌──────────────────────┐
                                                │Job: I  Pgm: I  Exec │
                                                └──────────────────────┘
```

There are no parameters for this command.

Example

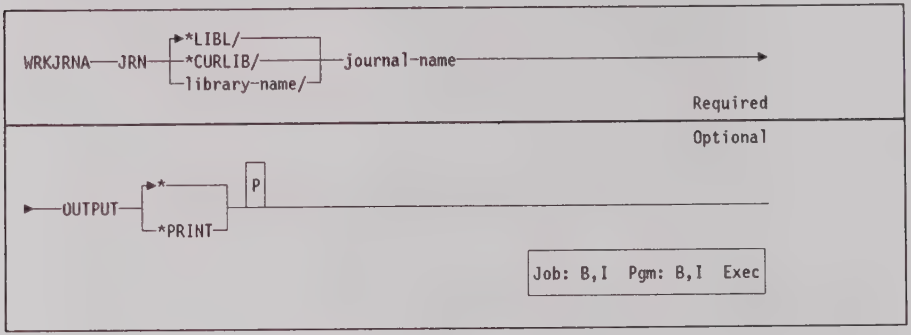
```
WRKJRN
```

This command displays the Work with Journal menu.

WRKJRNA (Work with Journal Attributes) Command

The Work with Journal Attributes (WRKJRNA) command displays the creation and operational attributes of a journal, including the names of the journal receivers currently attached to the journal. From the display that appears after the command is entered, options can be selected to display the names of all physical files currently being journaled, the names of all data base files currently having their access paths journaled, the receiver directory, and detailed information about a journal receiver. Receivers may also be deleted from the receiver directory.

If output is printed with the job's spooled printer output, then all of the information that is optionally displayed is printed (except for detailed information about journal receivers; for that information, use the Display Journal Receiver Attributes (DSPJRNRCVA) command).



JRN Parameter

Specifies the qualified name of the journal whose attributes are to be displayed (library-name/journal-name).

The possible library values are:

***LIBL:** The library list is used to locate the journal.

***CURLIB:** The current library for the job is used to locate the journal. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the journal is located.

OUTPUT Parameter

Specifies if the output from the command is displayed at the requesting work station or printed with the job's spooled printer output.

***:** Specifies, for an interactive job, that the output is displayed at the requesting work station. For a batch job, the output is printed with the job's spooled printer output.

***PRINT:** The output is printed with the job's spooled printer output.

Example

```
WRKJRNA   JRN(MYLIB/JRNLA)
```

This command displays the current journal attributes of JRNLA in library MYLIB.

WRKLIB (Work with Library) Command

The Work with Library (WRKLIB) command shows a list of names of libraries that the user has authority to. From that list several library updated functions can be performed.

Restrictions: (1) The user must have *USE authority for each library specified in order to display its contents. (2) To delete a specified library, you must have object existence authority to that library and *USE authority to the associated delete command for that object type.

```
                                                        Required
                  ┌►*LIBL──────────────────┐
                  ├─*CURLIB────────────────┤
                  ├─*USRLIBL───────────────┤ ┌─┐
                  │                        │ │P│
WRKLIB──LIB───────┼─*ALLUSR────────────────┼─┴─┴───────────────────────────────
                  ├─*ALL───────────────────┤
                  ├─library-name───────────┤
                  └─generic*-library-name──┘

                                                   ┌────────────────────┐
                                                   │Job: I  Pgm: I  Exec│
                                                   └────────────────────┘
```

LIB Parameter
Specifies the names of one or more libraries that are shown. If you do not have read authority for a specified library, that library is not shown.

***LIBL**: All the libraries in the user and system portions of the job's library list are shown.

***CURLIB**: The current library for the job is shown. If no current entry exists in the library list, QGPL is used.

***USRLIBL**: Only the libraries in the user portion of the job's library list are shown.

***ALLUSR**: All the nonsystem libraries, which include all user-defined libraries and the QGPL library, are shown. Libraries whose names start with the letter Q, other that QGPL, are not included.

***ALL**: All of the libraries in the system, including QSYS, are shown.

library-name: Only the library named is shown. The user must have *USE authority for the specified library.

generic-library*: Specify the generic library name that is shown. To specify the generic name, add an asterisk (*) after the last character in the generic name (for example, ABC*). If an * is not included with the name, the system assumes that the name is a complete library name.

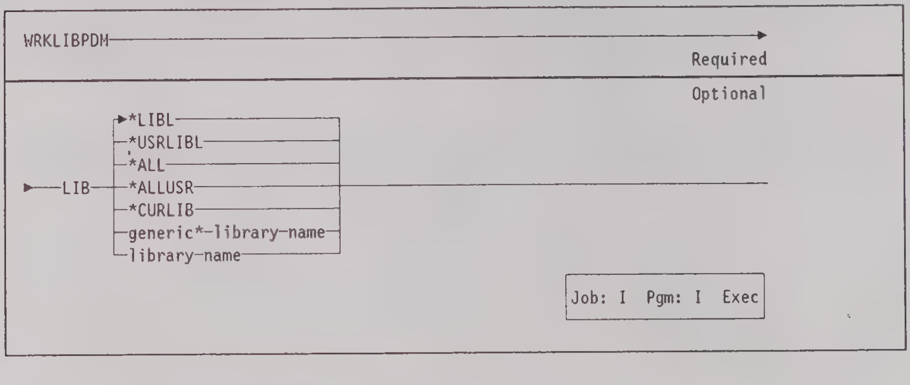
Example

```
WRKLIB   LIB(QJ*)
```

The command lists all libraries that begin with the letters QJ.

WRKLIBPDM (Work with Libraries Using PDM) Command

The Work with Libraries Using PDM (WRKLIBPDM) command allows you to work with lists of libraries. Using this command, you can bypass the Programming Development Manager (PDM) menu and the Specify Libraries to Work with display.



LIB Parameter
Specifies the libraries to work with this parameter to create a subset of a list of libraries by a generic library name.

The possible values are as follows:

- **<u>*LIBL:</u>** Specifies all the libraries in the user *and* system portions of the job's library list.
- ***USRLIBL:** Specifies only the libraries listed in the user portion of the job's library list.
- ***ALL:** Specifies all the libraries in the system, including QSYS and QTEMP.
- ***ALLUSR:** Specifies all the nonsystem libraries, which include *all* user-defined libraries and the QGPL library, not just those in the job's library list. Libraries other than QGPL whose names start with the letter Q are not included.
- ***CURLIB:** Specifies the current library in the library list. If you have not specified a current library, QGPL is used.

generic-library-name:* A partial library name qualified by an asterisk (*) displays a list of libraries that meet the specific criterion.

The generic name can be in one of the following formats:

- ABC* displays a list of all items whose name begins with the characters ABC. For example, ABC, ABCD, or ABCTEST.
- *ABC displays a list of all items whose name ends with the characters ABC. For example, ABC, DABC, or TESTABC.
- *B* displays a list of all items that have the character B anywhere in the name. For example, B, BALL, ABCD.
- A*C displays a list of all items that begin with the character A and end with the character C. For example, AC, ABC, or AZZZC.
- ″a*″ displays a list of all items that are quotes names that start with a. For example, ″a″, ″ab″, or ″ad″.
- **ALL displays a list of all items whose name ends with ALL. For example, ALL, BALL, or TESTALL. **ALL is a value needed for only these cases because *ALL is already used as a special value.

library-name: Specify a library name for a display with only that library name in the list.

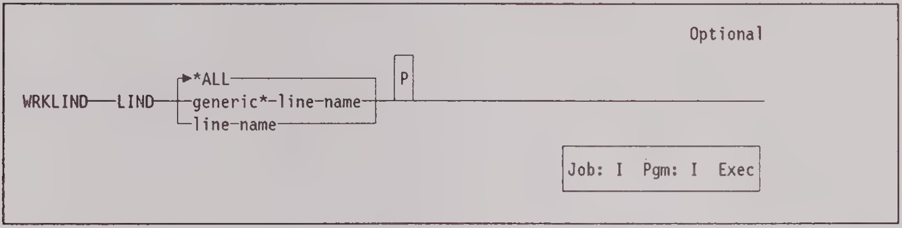
Example

```
WRKLIBPDM  LIB(ABC*)
```

This command shows the Work with Libraries using PDM Display and shows all libraries starting with ABC.

WRKLIND (Work with Line Descriptions) Command

The Work with Line Descriptions (WRKLIND) command provides an interactive interface to line description functions through the Work with Line Descriptions display. This command displays the Work with Line Descriptions display.



LIND Parameter
Specifies the line description(s) to work with. Specify which line descriptions to include in the list of line descriptions on the Work with Line Descriptions display.

***ALL:** You work with all lines.

generic-line-name:* Specify that you will work with lines whose names start with a specific character string (generic name).

line-name: Specify a specific line description name.

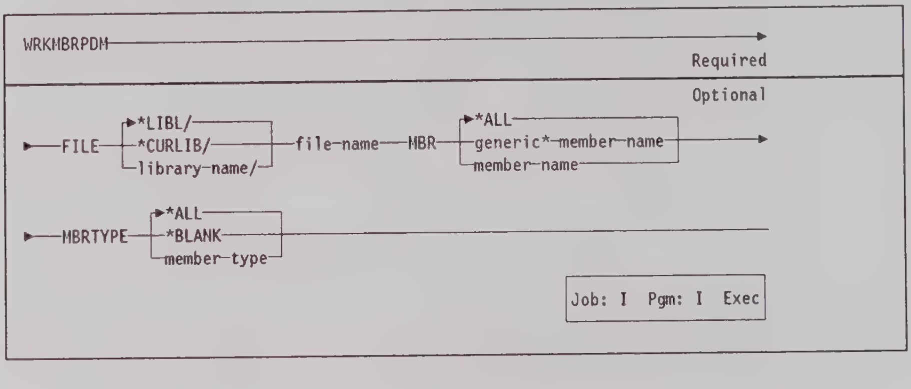
Example

```
WRKLIND  LIND(LINE01)
```

This command shows the Work with Line Descriptions display with an entry for line 'LINE01'. If LINE01 does not exist, the list will be blank (no entries will be shown).

WRKMBRPDM (Work with Members Using PDM) Command

The Work with Members Using PDM (WRKMBRPDM) command allows you to work with lists of members. Using this command, you can bypass the Programming Development Manager (PDM) menu and the Specify Members to Work with display.



FILE Parameter
Specifies the qualified name of the file (in the form LIB/FILE) that contains the members you want to work with. This is a required parameter, and no default is supplied for the file name. The file can be a source physical file or a data physical file.

The possible library values are:

***LIBL:** The library list is used to locate the file.

***CURLIB:** The current library for the job is used to locate the file. If no current library entry exists in the library list, QGPL is used.

library-name: The specified library name is used to locate the file.

MBR Parameter
Specifies the member name. You can use this parameter to work with all the members or a subset of members in the specified file.

***ALL:** Specify *ALL for a list of all the members in the specified file.

generic-member-name:* Specify a partial member name qualified by an asterisk (*) to display a list of members that meet the specific criterion.

The generic name can be in one of the following formats:

- ABC* displays a list of all items whose name begins with the characters ABC. For example, ABC, ABCD, or ABCTEST.
- *ABC displays a list of all items whose name ends with the characters ABC. For example, ABC, DABC, or TESTABC.
- *B* displays a list of all items that have the character B anywhere in the name. For example, B, BALL, ABCD.
- A*C displays a list of all items that begin with the character A and end with the character C. For example, AC, ABC, or AZZZC.
- "a*" displays a list of all items that are quoted names that start with a. For example, "a", "ab", or "ad".
- **ALL displays a list of all items whose name ends with ALL. For example, ALL, BALL, or TESTALL. **ALL is a value needed for only these cases because *ALL is already used as a special value.

member-name: Specify a member name for a display with only that member name in the list.

MBRTYPE Parameter
Specifies the member type. You can use this parameter to work with all member types in a specified file or a subset of members.

***ALL:** Specify *ALL to display a list of all members with any member type.

***BLANK:** Specify *BLANK to display a list of all members with no type.

member-type: Specify any member type to display a list of all members of that particular type.

Example

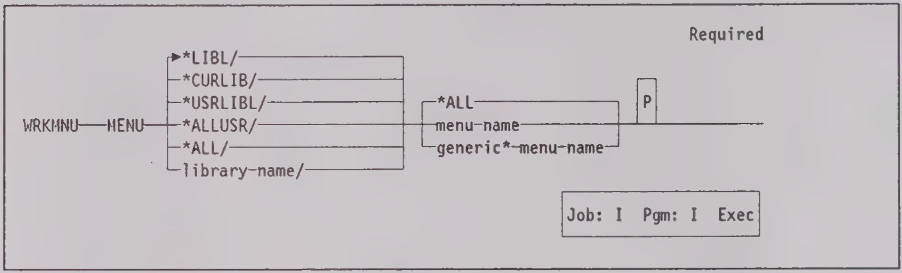
```
WRKMBRPDM  FILE(MYLIB/QDDSSRC)  MBR(*ALL)  MBRTYPE(PLI)
```

This command shows the Work with Members Using PDM display and lists all members in file QDDSSRC in library MYLIB that are of type PLI.

WRKMNU (Work with Menus) Command

The Work with Menus (WRKMNU) command shows a list of menus and allows the user to perform several menu related functions including display the menu or menu attributes.

Restriction: The user must have authority to both the menu and the library where it is located for the operation he wants to perform.



MENU Parameter

Specifies the qualified name of the menus to be shown on the Work with Menus display. A specific menu or a generic menu can be specified; either type can be optionally qualified by a library name.

The possible library values are:

- ***LIBL**: All libraries in the user and system portions of the job's library list are searched.
- ***CURLIB**: The current library for the job is searched. If no current library exists in the library list, QGPL is used.
- ***USRLIBL**: Only the libraries listed in the user portion of the library list are searched.
- ***ALLUSR**: All nonsystem libraries, including all user-defined libraries and the QGPL library, not just those in the job's library list, are searched. Libraries whose names begin with the letter Q, other than QGPL, are not included.
- ***ALL**: All libraries in the system, including QSYS, are searched.
- *library-name*: Only the library named in this parameter is searched.

***ALL:** All menus (in the library(s) specified) are listed on the Work with Menus display.

menu-name: Specify the name of the menu being listed.

generic-menu-name:* Specify the generic name of the menu being listed. A generic name can be specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified for the MENU parameter, then all menus run the library(s) specified that have names with the same prefix as the generic menu name are listed.

For more information on the use of generic functions, refer to Appendix G, "Generic Functions."

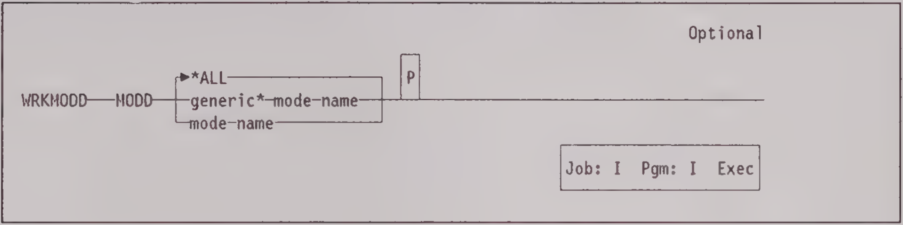
Example

```
WRKMNU PERSLIB/OE*
```

This command shows a list of all menus in library PERSLIB whose names begin with OE. From the list shown, the user can display the menu or the menu attributes.

WRKMODD (Work with Mode Descriptions) Command

The Work with Mode Descriptions (WRKMODD) command provides an interactive interface to mode description functions through the Work with Mode Descriptions display. This command displays the Work with Mode Descriptions display.



MODD Parameter
Specifies the mode description(s) you work with. Specify which mode description(s) to include in the list of mode descriptions on the Work with Mode Descriptions display.

***ALL:** You work with all mode descriptions.

generic-mode-name:* Specify that you will work with mode descriptions whose names start with a certain character string (generic name).

mode-name: Specify a specific mode description.

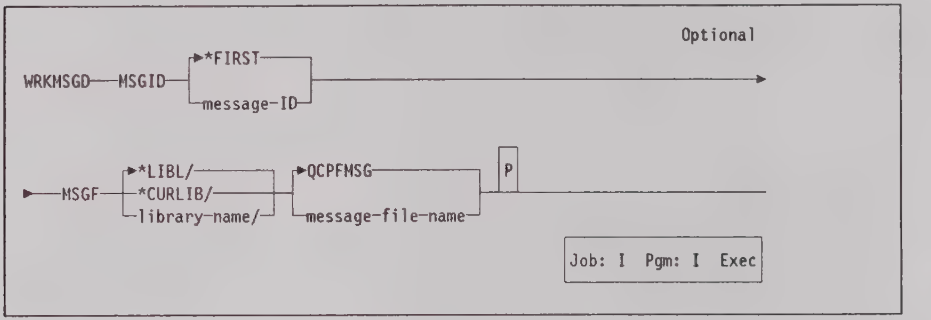
Example

```
WRKMODD  MODD(*ALL)
```

This command shows the Work with Mode Descriptions display with entries for all existing mode descriptions.

WRKMSGD (Work with Message Descriptions) Command

The Work with Message Descriptions (WRKMSGD) command shows detailed information about the messages contained in a message file. This command can be used to add, change, and remove message descriptions through the Work with Message Descriptions display. When message descriptions are being changed from this display, the current values for the message are shown in the command prompt. However, there is a 512-character limit for the second-level message.



MSGID Parameter

Specifies the message identifier with which to begin showing a list of message descriptions in the message file specified by the MSGF parameter.

***FIRST:** The first message description in the message file is used to begin the list display.

message-ID: Specify the message identifier used to begin the list display. The message identifiers must each be 7 characters long and in the following format:

`pppmmnn`

The first 3 characters must be a code consisting of an alphabetic character followed by 2 alphameric (alphabetic or decimal) characters; the last 4 characters may consist of the decimal numbers 0 through 9 and the characters A through F.

Note: Message identifiers using the MCH code (MCHnnnn) use only the numbers 0 through 9 in the last 4 characters.

MSGF Parameter

Specifies the qualified name of the message file from which the message descriptions are taken.

The possible library values are:

***LIBL:** The library list is used to locate the message file.

***CURLIB:** The current library for the job is used to locate the message file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the message file is located.

QCPFMSG: Message descriptions are taken from the system message file, QCPFMSG.

message-file-name: Specify the qualified name of the message file from which the message descriptions are taken.

Example

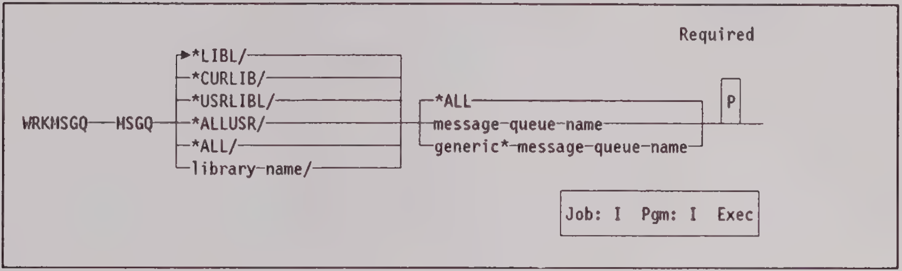
```
WRKMSGD   MSGF(QSYS/QCPFMSG)
```

This command shows the Work with Message Descriptions display for message descriptions found in message file QCPFMSG in library QSYS. From the display, the user can change, delete, display or print details of these message descriptions.

WRKMSGQ (Work with Message Queues) Command

The Work with Message Queues (WRKMSGQ) command shows a list of message queues and allows the user to display, change, delete, and clear specified message queues.

Restriction: The user must have authority to both the message queue and the library where it is located for the operation he wants to perform.



MSGQ Parameter

Specifies the qualified name of the message queues to be shown on the Work with Message Queues display. A specific message queue or a generic message queue can be specified; either type can be optionally qualified by a library name.

The possible library values are:

- ***LIBL**: All libraries in the user and system portions of the job's library list are searched.
- ***CURLIB**: The current library for the job is searched. If no current library exists in the library list, QGPL is used.
- ***USRLIBL**: Only the libraries listed in the user portion of the library list are searched.
- ***ALLUSR**: All nonsystem libraries, including all user-defined libraries and the QGPL library are searched. Libraries whose names begin with the letter Q, other than QGPL, are not included.
- ***ALL**: All libraries in the system, including QSYS, are searched.
- *library-name*: Only the library named in this parameter is searched.

***ALL:** All message queues (in the library specified) are listed on the Work with Message Queues display.

message-queue-name: Specify the name of the message queue to be listed.

generic-message-queue-name:* Specify the generic name of the message queue(s) to be listed. A generic name can be specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified for the MSGQ parameter, then all message queues in the specified library that have names with the same prefix as the generic message queue name are listed.

For more information on the use of generic functions, refer to Appendix G, "Generic Functions."

Example

```
WRKMSGQ PERSLIB/MQ*
```

This command shows a list of all message queues in library PERSLIB whose names begin with MQ. From the list shown, the user can display, change, delete, or clear any or all of the message queues shown.

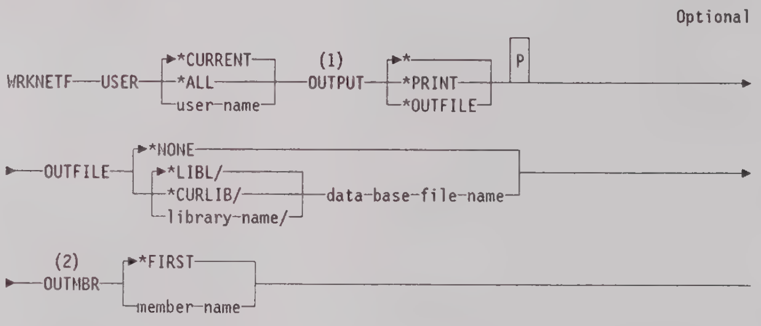
WRKNETF (Work with Network Files) Command

The Work with Network Files (WRKNETF) command displays or prints a list of files that have arrived for a user, or creates an output file containing a list of the files.

If the list is displayed, you can enter an option to select a function to be performed on the file. You can:

- RCVNETF (receive the file into a user file).
- DLTNETF (delete the file).
- Browse (browse the file; not valid for save files).
- Submit files (submit the job stream).

Restrictions: (1) A user with security officer authority can display the network files for any user. A user other than the security officer can display only those files that were sent to him or to his group profile. (2) To perform any of the options from this display, you must be authorized to the command corresponding to that option. For example, you must be authorized to the Browse Physical File Member (BRWPFM) command for the browse function, and the Submit Data Base Jobs (SBMDBJOB) command for the submit job function.



(1) OUTPUT(*OUTFILE) can be specified only if a file name is specified for OUTFILE.
(2) A member name can be specified for OUTMBR only if a file name is specified for OUTFILE.

Job: B,I Pgm: B,I Exec

USER Parameter
Specifies the user for whom the files are displayed.

*CURRENT: The network files for the current user are displayed.

***ALL:** The network files for all users are displayed. Only a user with security officer rights can specify this value.

user-name: Specify the name of the specified user whose files are displayed. Only a user with security officer rights can specify a name other than that of his own or his group profile.

OUTPUT Parameter
Specifies if the output from the command is displayed at the requesting work station, listed with the job's spooled output on a printer, or sent to a data base file.

_*:_ The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

***PRINT:** The output is listed with the job's spooled output on a printer.

***OUTFILE:** The output is not printed but is directed to the data base file specified in the OUTFILE parameter.

OUTFILE Parameter
Specifies the name of the data base file to which the output of the display is directed. If the specified data base file does not exist, this command creates it in the specified library.

Note: The outfile format must be the same as QNFDNTF of the system file QANFDNTF.

*NONE: The output is not directed to a data base file.

The possible library values are:

***LIBL:** The library list is used to locate the date file.

***CURLIB:** The current library for the job is used to locate the data file. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the data base is located.

data-base-file-name: Specify the name of the data base file that receives the output of the display.

OUTMBR Parameter

Specifies the data base file member to which the output of the display is directed. If a member already exists, the system clears it, and records are added to the file. If a member does not exist and a member name is not specified, the system creates a member with the same name as the file specified in the OUTFILE parameter. If a member name is specified and the member does not exist, the system creates a member with the specified name.

***FIRST:** The first member in the file receives the output.

member-name: Specify the name of the member in the file that receives the output.

Examples

```
WRKNETF
```

This command displays all network files for the user running this command. If the command is issued as an interactive job, the list of files is displayed at the requesting work station. If the command is issued as a batch job, the list of files is printed with the job's spooled output.

```
WRKNETF USER(USR1) OUTPUT(*PRINT)
```

This command lists the network files for USR1 and prints the output with the job's spooled output. This command can only be issued by USR1, a member of the USR1 group, or a user with security officer rights.

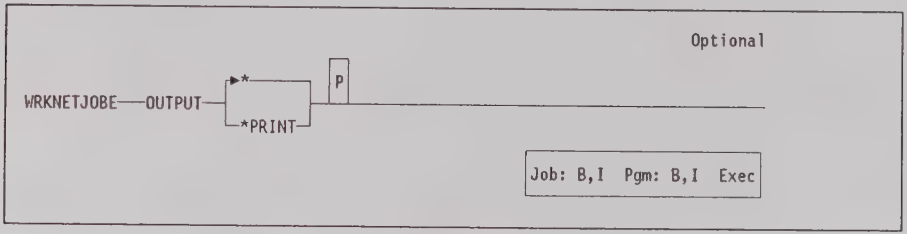
```
WRKNETF USER(*ALL) OUTPUT(*OUTFILE) OUTFILE(NETFILES)
```

This command lists the network files for all users and is written to the first member of a data base named NETFILES. If the file exists in a library on the library list, the existing file is used; otherwise, the file is created in library QGPL. If the file did not exist, or did not contain any members, a member with the same name as the file is added to the file; otherwise, the first member of the file is cleared and used. This command can be issued only by a user with security officer rights.

WRKNETJOBE (Work with Network Job Entry) Command

The Work with Network Job Entry (WRKNETJOBE) command displays the network job entries. There is one entry for each user or distribution group who may submit jobs to this system. This entry is used to determine whether the job stream is automatically submitted, placed on the queue of network files for a user, or rejected. This entry also specifies the user profile that is used for checking the authority to the job description referenced in the batch job.

Restrictions: (1) Any user can display the network job entries with this command. (2) You must be explicitly authorized to the commands corresponding to the options that can be selected from this display in order to select these options.



OUTPUT Parameter
Specifies if the output is displayed at the requesting work station or listed with the job's spooled output on a printer.

*: The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

*PRINT: The output is listed with the job's spooled output on a printer.

Examples

```
WRKNETJOBE OUTPUT(*PRINT)
```

This command prints the network job entries with the job's spooled output.

```
WRKNETJOBE OUTPUT(*)
```

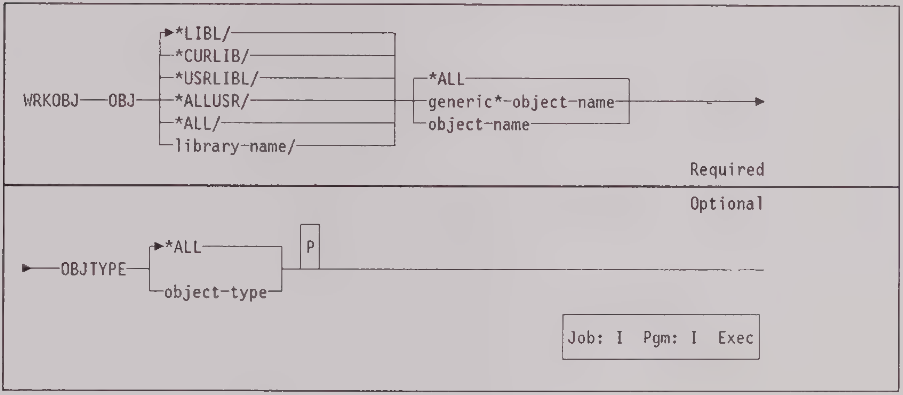
This command, if issued in an interactive job, displays the network job entries at the requesting work station. If the command is issued in a batch job, the network job entries are printed with the job's spooled output.

WRKOBJ (Work with Object) Command

The Work with Object (WRKOBJ) command shows a list of names and attributes of specified objects in specified libraries.

Only the object attributes of each object are shown; the data attributes of data in the object, and the actual data in the object, are not shown. Also, if an object being shown has been damaged (possible by a system failure), the fact that it is damaged is indicated on the display.

Restriction: The user either must own the objects or must have *USE authority for the specified libraries and some authority (meaning any one of the object authority) for each of the objects. If he does not have read authority for a library, none of its objects are shown.



OBJ Parameter

Specifies which objects in the library or libraries are shown. If no library qualifier is specified, *LIBL is assumed and all libraries in the user portion of the job's libraries are searched for the objects.

Depending on the library qualifier specified or assumed, the following libraries (for which the user has the authority) are searched for the specified objects:

- *LIBL: All the libraries in the user and system portions of the job's library list are searched.
- *CURLIB: The current library for the job is searched. If no current entry exists in the library list, QGPL is used.
- *USRLIBL: Only the libraries listed in the user portion of the job's library list are searched.
- *ALLUSR: All the nonsystem libraries, which include all user-defined libraries and the QGPL library, that are not in the job's library list are searched. Libraries whose names start with the letter Q, other then QGPL, are not included.
- *ALL: All of the libraries in the system, including QSYS, are searched.
- *library-name*: Only the library named is searched. The user must have *USE authority for the specified library.
- *generic*-library-name*: Specifies the generic library names that are searched.

Only those files that the user has some authority to are shown.

***ALL:** All objects in the libraries identified in the library qualifier that are of the types specified by the OBJTYPE parameter are shown. The user can display only those objects for which he has some authority.

generic-object-name: Specify the generic object name that is shown. To specify the generic name, add an asterisk (*) after the last character in the generic name (for example, ABC*). If an * is not included with the name, the system assumes that the name is a complete object name.

object-name: Specify the qualified name of the object that is shown. If the library qualifier is *ALL, *ALLUSR, or a library name, all objects of the specified types that the user has some authority to (for example, *USE authority) and that are in the specified libraries are shown. If the qualifier is *USRLIBL or *LIBL, then only one object type (not *ALL) can be specified, and only the first object found is shown.

OBJTYPE Parameter

Specifies which types of objects are shown. For a list of object types, see the OBJTYPE parameter in Appendix A, "Expanded Parameter Descriptions."

***ALL:** All object types are shown that have the specified object name.

object-type: Specify one or more values (up to a maximum of 26) for the types of object that can be shown. Any of the object types can be specified; refer to the charts in the expanded description of the OBJTYPE parameter in Appendix A, "Expanded Parameter Descriptions." All objects in the specified libraries, as well as libraries themselves, that have the object types specified are shown. If the library qualifier specified or assumed for the OBJ parameter is *USRLIBL or *LIBL, then only one object type (not *ALL) can be specified here.

Example

```
WRKOBJ   OBJ(X/PAY) OBJTYPE(*ALL)
```

The object or objects (for which the user has some authority) that is named PAY and is located in the library X is shown.

WRKOBJOWN (Work with Objects by Owner) Command

The Work with Objects by Owner (WRKOBJOWN) command is used to manage objects for any user profile. WRKOBJOWN allows the user to display objects owned by a user profile and perform the following:

- Edit object authority
- Delete the object
- Display object authority
- Display the object's description
- Change ownership of the object

Restriction: The user must have object management authority to the profile being displayed.

```
                        ┌►*CURRENT──────────┐
WRKOBJOWN───USRPRF──────┤                   ├──────────────────────────────
                        └─user-profile-name─┘

                                                  ┌──────────────────────┐
                                                  │Job: I  Pgm: I  Exec  │
                                                  └──────────────────────┘
```

USRPRF Parameter
Specifies the name of the user profile whose objects you want to work with.

<u>*CURRENT</u>: The current user profile of the job is used.

user-profile-name: Specify the name of the user profile you want to work with.

Example

```
WRKOBJOWN  USRPRF(*CURRENT)
```

This command displays the objects owned by the user running this command.

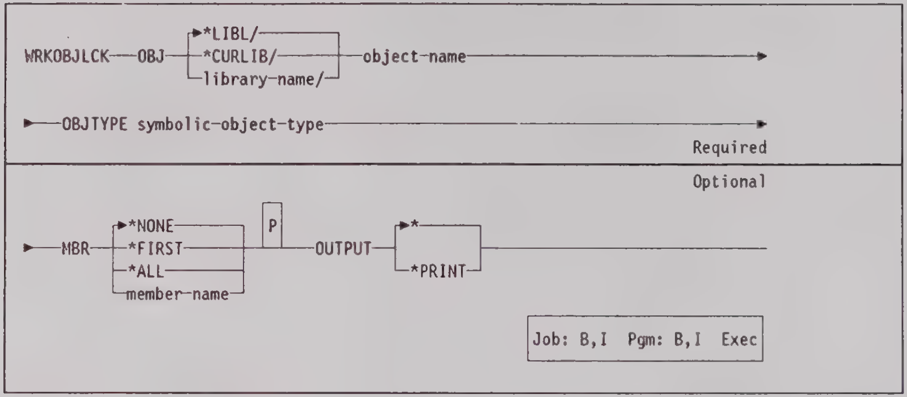
```
WRKOBJOWN  USRPRF(BARTH)
```

This command displays the objects owned by BARTH user profile.

WRKOBJLCK (Work with Object Lock) Command

The Work with Object Lock (WRKOBJLCK) command allows you to work with or change the object lock requests in the system for a specified object. You can work with both held locks and locks being waited for.

Note: This command does not show record locks for data base files.



OBJ Parameter

Specifies the qualified name of the operating system object for which locks are being displayed.

The possible library values are:

***LIBL:** The library list is used to locate the operating system object.

***CURLIB:** The current library for the job is used to locate the operating system object. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the operating system object is located.

If a file member is specified for a file, and the file's library value is *LIBL, the first occurrence of the file in the job's library search list is searched for the member. For object types that exist only in library QSYS (for example, *DEVD), QSYS and *LIBL are the only library names that are accepted.

OBJTYPE Parameter

Specifies the object type of the operating system object for which locks are being displayed. Enter the predefined value that identifies the object type.

MBR Parameter

Specifies the member name of an operating system data base file. This parameter is valid only when a data base file has been specified on the OBJ parameter.

***NONE:** No member locks are displayed, but file level locks are displayed. (The display of member locks for all the members in the file may be requested from the File Locks display.)

***FIRST:** Specifies that the first member in the named file is displayed.

***ALL:** Specifies that member locks for all the members in the file are displayed.

member name: Specify the name of the data base file member for which locks are displayed.

OUTPUT Parameter

Specifies if the display is shown that allows you to work with or change object lock displayed at the requesting work station or printed with the job's spooled output. (Refer to Appendix A, "Expanded Parameter Descriptions" for an expanded description of the OUTPUT parameter, and to Appendix D, "Files Used by CL Commands" for the name of the printer file used by this command.)

***:** The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

***PRINT:** The output is printed with the job's spooled output.

Example

```
WRKOBJLCK   OBJ(*LIBL/LOCKEDFILE)  OBJTYPE(*FILE)  +
  MBR(LOCKEDMBR)  OUTPUT(*PRINT)
```

This command allows you to work with or change the lock information for member LOCKEDMBR of file LOCKEDFILE being written to the job's spooled printer file.

WRKOBJPDM (Work with Objects Using PDM) Command

The Work with Objects Using PDM (WRKOBJPDM) command allows you to work with lists of objects. Using this command, you can bypass the Programming Development Manager (PDM) menu and the Specify Objects to Work with display.

```
WRKOBJPDM──────────────────────────────────────────────────────────────►
                                                                Required
────────────────────────────────────────────────────────────────────────
                                                                Optional
        ┌►*CURLIB/──────┐        ┌►*ALL─────────────────┐
►──LIB──┤               ├──OBJ───┼─generic*-object-name─┼──────────────►
        └─library-name/─┘        └─object-name──────────┘

           ┌►*ALL─────────┐           ┌►*ALL─────────────┐
►──OBJTYPE─┤              ├──OBJATR───┼─*BLANK───────────┼──────────────
           └─object-type──┘           └─object-attribute─┘
                                                     ┌───────────────────┐
                                                     │Job: I  Pgm: I Exec│
                                                     └───────────────────┘
```

LIB Parameter
Specifies the library that contains the objects you want to work with.

The possible library values are:

***CURLIB:** The current library for the job is used to locate the objects. If no current library entry exists in the library list, QGPL is used.

library-name: The library specified is used to locate the objects.

OBJ Parameter
Specifies the object name. You can use this parameter to work with all the objects or a subset of objects in the specified library.

***ALL:** Specify *ALL for a list of all the objects in the specified library.

generic-name: Specify a partial object name qualified by an asterisk (*) to display a list of objects that meet the specific criterion.

The generic name can be in one of the following formats:

- ABC* displays a list of all items whose name begins with the characters ABC. For example, ABC, ABCD, or ABCTEST.
- *ABC displays a list of all items whose name ends with the characters ABC. For example, ABC, DABC, or TESTABC.
- *B* displays a list of all items that have the character B anywhere in the name. For example, B, BALL, ABCD.
- A*C displays a list of all items that begin with the character A and end with the character C. For example, AC, ABC, or AZZZC.
- ″a*″ displays a list of all items that are quoted names that start with a. For example, ″a″, ″ab″, or ″ad″.
- **ALL displays a list of all items whose name ends with ALL. For example, ALL, BALL, or TESTALL. **ALL is a value needed for only these cases because *ALL is already used as a special value.

object-name: Specify an object name for a list of all the objects with that name.

OBJTYPE Parameter
Specifies the object type. You can use this parameter to work with all object types or a subset of objects.

***ALL:** Specify *ALL to display a list of all objects regardless of the object type.

object-type: Specify any valid system object type to display a list of all objects of that particular type.

OBJATR Parameter
Specifies the object attribute. You can use this parameter to work with all object attributes or only specific object attributes.

***ALL:** Specify *ALL to display a list of all objects regardless of the object attribute.

***BLANK:** Specify a list of all objects with no attribute.

object-attribute: Specify any object attribute to display a list of all objects with that particular attribute. If you specify the object attribute, you do not have to specify the object type.

Example

```
WRKOBJPDM  LIB(TESTLIB)  OBJ(*TEST*) OBJTYPE(*FILE)  OBJATR(PHY)
```

This command shows the Work with Objects Using PDM display and shows all physical files in library TESTLIB with 'TEST' anywhere in the name.

WRKOUTQ (Work with Output Queue) Command

The Work with Output Queue (WRKOUTQ) command allows you to work with the overall status of all output queues or the detailed status of a specific output queue. The status of the queues may change while the command is being run.

```
                                                                  Optional
                       ┌►*ALL──────────────────────────────────┐
WRKOUTQ──OUTQ──┤ ┌►*LIBL/────────┐                      ├──────────────────►
               └─┼─*CURLIB/──────┼─output-queue-name─┘
                 └─library-name/─┘

                   ┌►*──────┐  ┌─┐
►──OUTPUT──┤        │  │P│
           └─*PRINT─┘──└─┘─────────────────────────────────────────────

                                                ┌─────────────────────────┐
                                                │Job: B,I  Pgm: B,I  Exec │
                                                └─────────────────────────┘
```

OUTQ Parameter

Specifies that all output queues are shown, or specifies the qualified name of the output queue that is to have its status shown.

***ALL:** The overall status of all output queues is shown.

The possible library values are:

***LIBL:** The library list is used to locate the output queue.

***CURLIB:** The current library for the job is used to locate the output queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the output queue is located.

output-queue-name: Specify the qualified name of the output queue whose detailed status information is shown.

OUTPUT Parameter

Specifies whether the output from the command is shown at the requesting work station or is printed with the job's spooled output on a printer. (Refer to Appendix A, "Expanded Parameter Descriptions" in Volume 1 for an expanded description of the OUTPUT parameter, and to Appendix D, "Files Used by CL Commands" in Volume 1 for the name of the printer file used by this command.)

***:** The output is shown (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

***PRINT:** The output is printed with the job's spooled output on a printer.

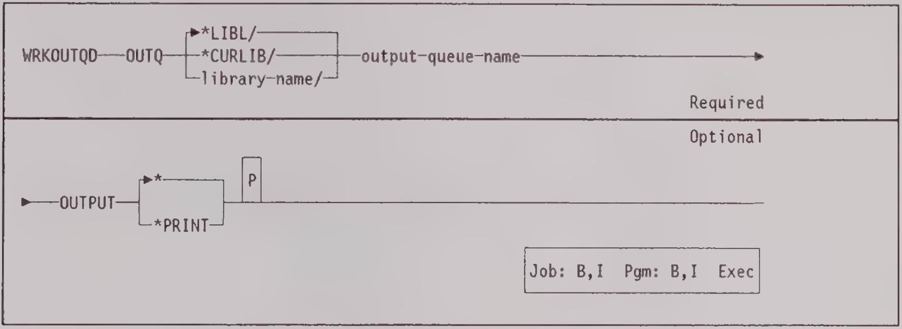
Example

```
WRKOUTQ  OUTQ(QGPL/QPRINT)
```

This command shows the detailed status information about the output queue named QPRINT in the library named QGPL. Each spooled file on the QPRINT output queue is shown.

WRKOUTQD (Work with Output Queue Description) Command

The Work with Output Queue Description (WRKOUTQD) command allows you to work with the description of the specified output queue. The description of the queue may change while the command is being run.



OUTQ Parameter

Specifies the name of the output queue whose description is shown.

The possible library values are:

<u>*LIBL:</u> The library list is used to locate the output queue.

***CURLIB:** The current library for the job is used to locate the output queue. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the output queue is located.

OUTPUT Parameter

Specifies whether the output from the command is shown at the requesting work station or printed with the job's spooled output on a printer. (Refer to Appendix A, "Expanded Parameter Descriptions" in Volume 1 for an expanded description of the OUTPUT parameter, and to Appendix D, "Files Used by CL Commands" in Volume 1 for the name of the printer file used by this command.)

<u>*:</u> The output is shown (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

***PRINT:** The output is printed with the job's spooled output on a printer.

Example

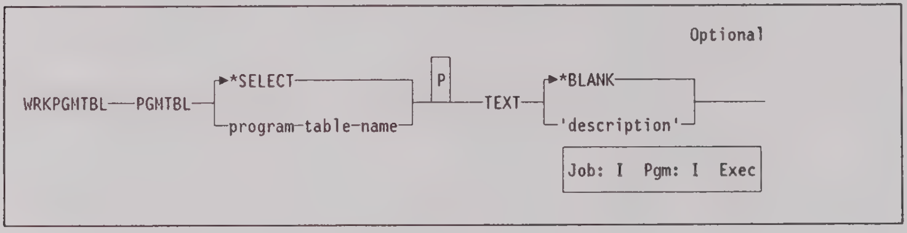
```
WRKOUTQD  OUTQ(QGPL/QPRINT)
```

This command shows the descriptive information for the output queue named QPRINT in the library named QGPL.

WRKPGMTBL (Work with Program Table) Command

The Work with Program Table (WRKPGMTBL) command lets you create finance program tables and, once they are created, lets you add or delete program names in these tables. Several finance program tables can be defined, but you must use unique names for each table. If the program table you specify on this command does not exist, it is created.

Program table updates can be accessed by any finance job that is submitted after all changes are completed.



PGMTBL Parameter

Specifies the name of a table that contains finance program IDs and user-associated program names.

*SELECT: Displays the list of existing program tables. With this display you can add new program tables to the system, delete old program tables from the system, or change the contents of existing program tables.

program-table-name: Specify the name of the program table that you want to work with.

TEXT Parameter

Specifies the user-defined text that describes the program table. (For an expanded description of the TEXT parameter, see Appendix A, "Expanded Parameter Descriptions.")

*BLANK: No text is specified.

'description': Specify no more than 50 characters, enclosed in apostrophes. If you enter text in this field and specify PGMTBL(*SELECT), the text you enter here is shown on the Select Program Table display.

Examples

```
WRKPGMTBL PGMTBL(*SELECT)
```

This command shows the names of all finance program tables. You can add a new table, select an existing table for update, or remove tables on the display.

```
WRKPGMTBL PGMTBL(PGMTBL1) TEXT('Program group one')
```

This command shows the program table PGMTBL1, allowing you to add, change, or remove table entries. If PGMTBL1 does not exist when the WRKPGMTBL command is entered, a new program table is created using the specified text.

WRKPRB (Work with Problem) Command

The Work with Problem (WRKPRB) command shows the Work with Problem display. You can use this command to work with problems that were detected by the system or detected by the user.

Problems can be sorted two ways:

- By the status of the problem
- By the date they were entered in the log

The four types of status are:

Opened	The problem was identified; problem analysis has not been run.
Ready	Problem analysis was run; the problem is ready to be prepared.
Prepared	Information relating to a service request was added.
Sent	The problem was sent to service support.

Problems can be deleted from the log using the Delete Problem (DLTPRB) command.

```
                                                        Optional
WRKPRB----------------------------------------------------------

                                          Job: I   Pgm: I   Exec
```

There are no parameters for this command.

Example

```
WRKPRB
```

This command shows the Work with Problem Log display.

WRKPRDINF (Work with Product Information) Command

The Work with Product Information (WRKPRDINF) command is used to work with information that helps you contact, or be contacted by, various support centers. The contact information is supplied to you by your market support center and your service support center. This command shows a menu where you can select one of the four kinds of support functions.

```
                                                      Required
WRKPRDINF─────────────────────────────────────────────────────
                                              ┌────────────────────────┐
                                              │Job: I  Pgm: I  Exec    │
                                              └────────────────────────┘
```

There are no parameters for this command.

Example

```
WRKPRDINF
```

This command displays the IBM Information Network Signon.

WRKQRY (Work with Query) Command

The Work with Query (WRKQRY) command shows the Work with Queries display that allows you to create, change, copy, delete, display, and print a query definition and to run a query.

```
WRKQRY——————————————————————————————————————————————————————

                                              ┌──────────────┐
                                              │Job: I  Pgm: I│
                                              └──────────────┘
```

There are no parameters for this command.

Example

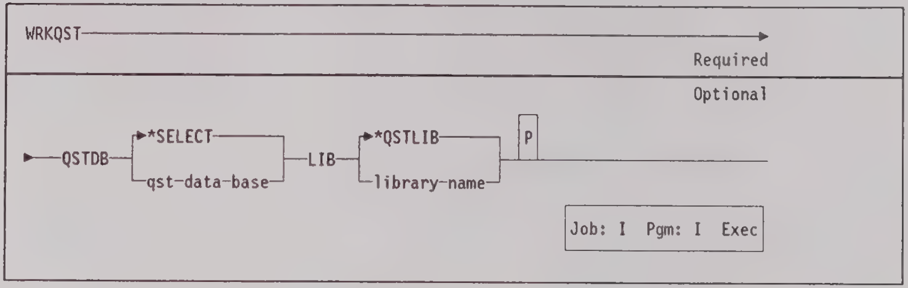
`WRKQRY`

This command shows the Work with Queries display.

WRKQST (Work with Question) Command

The Work with Question (WRKQST) command is used to review questions that the user has asked before.

Restriction: This command only runs interactively.



QSTDB Parameter

Specify the Question-and-Answer (Q & A) data base in which you want to work.

*SELECT: You will be prompted to enter or select a Q & A data base. If only one Q & A data base exists on your system, it will be selected for you.

qst-data-base: Specify the Q & A data base that you want to use.

LIB Parameter

Specifies the name of the library that contains the Q & A data base.

*QSTLIB: The library containing the specified Q & A data base used.

library-name: Specify the library where the Q & A data base is located.

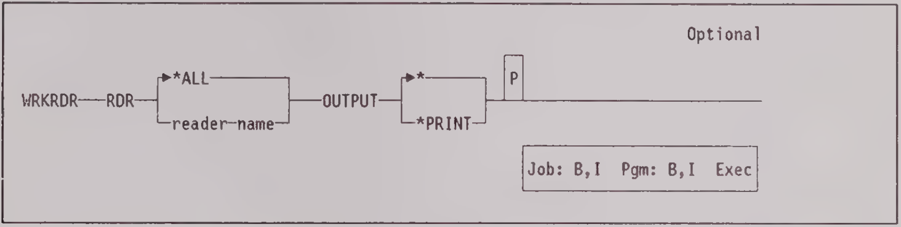
Example

```
WRKQST
```

This command shows the Work with Questions You Asked display.

WRKRDR (Work with Readers) Command

The Work with Readers (WRKRDR) command allows you to work with the overall status of all readers or the detailed status of a specific reader. The status of the readers may change while the command is in process.



RDR Parameter
Specifies whether overall status is given for all readers or detailed status is given for a specified reader.

*ALL: The overall status of all readers is displayed.

reader-name: Specify the name of the reader whose detailed information is displayed.

OUTPUT Parameter
Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output on a printer. (Refer to Appendix A, "Expanded Parameter Descriptions" in Volume 1 for an expanded description of the OUTPUT parameter, and to Appendix D, "Files Used by CL Commands" in Volume 1 for the name of the printer file used by this command.)

_*:_ The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

***PRINT:** The output is printed with the job's spooled output on a printer.

Examples

```
WRKRDR
```

This command shows the names of all readers, their types, device files, and status.

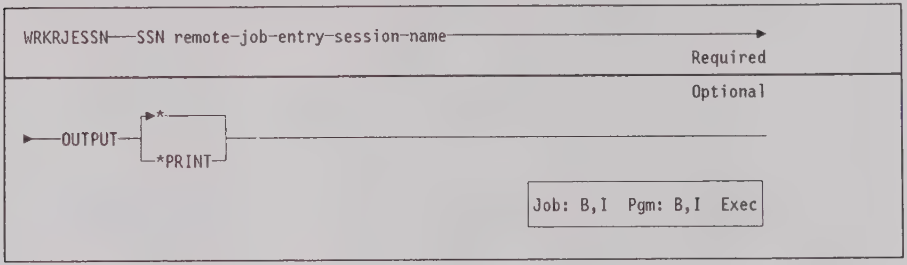
```
WRKRDR  RDR(DISKREAD)
```

This command shows detailed information about the reader DISKREAD.

WRKRJESSN (Work with RJE Session) Command

The Work with RJE Session (WRKRJESSN) command shows information about an active Remote Job Entry (RJE) session status. You can start or cancel readers and writers, and show information about active readers and writers. You can show information about the RJE session used by an active reader or writer, and about the session attributes.

Restriction: To use this command, you must have object operational authority to the session description and read authority to the library where the session description is stored.



SSN Parameter
Specifies the name of the RJE session that is shown.

OUTPUT Parameter
Specifies if the output from the command is shown at the requesting work station or printed with the spooled output of the job.

***:** The output is shown at the work station if it is requested by an interactive job. The output is printed with the spooled output of the job if it is requested by a batch job.

***PRINT:** The output is printed with the spooled output of the job. The IBM-supplied device file QPRJESTS prints the RJE session status.

Example

```
WRKRJESSN  SSN(RJE)  OUTPUT(*)
```

This command sends a list of detailed information of RJE devices associated with the active RJE session to the spooling queue of the job for printing. If the command is entered from an interactive job, a menu of RJE devices is displayed where selections are made.

WRKRPYLE (Work with System Reply List Entry) Command

The Work with System Reply List Entry (WRKRPYLE) command can be used to display or print all the reply entries currently in the system message reply list. The system reply list contains replies that are to be automatically sent in response to inquiry messages.

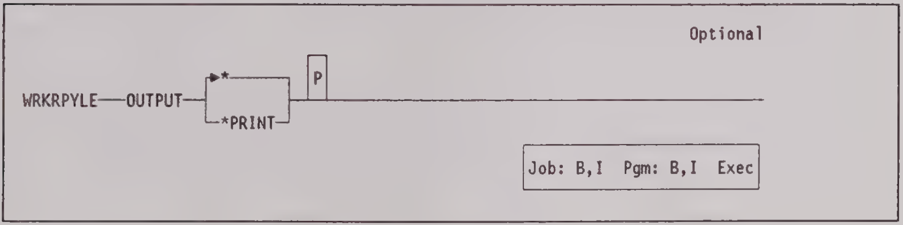
The following information is shown for each reply in the system reply list:

- Sequence number
- Message ID
- Reply
- Dump indication
- Compare value
- Compare start position

From this display you can add, change, or remove individual reply list entries. The reply list is used as a source for automatic responses to predefined inquiry messages.

The reply list is only used when an inquiry message is sent by a job that has the inquiry message reply attribute of the system reply list specified (INQMSGRPY (*SYSRPYL) is specified).

You can add reply list entries with the Add Reply List Entry (ADDRPYLE) command; specific attributes of a reply list entry can be changed with the Change Reply List Entry (CHGRPYLE) command. Each reply list entry remains in the list until it is removed by the Remove Reply List Entry (RMVRPYLE) command.



OUTPUT Parameter
Specifies whether the reply list entries are printed or displayed. If the job is a batch job or if *PRINT is specified, the system reply list is listed in the output file QPRTRPYL.

***:** Specifies that the entries are displayed, unless the job is in batch mode, in which case the reply into entries are printed.

***PRINT:** Specifies that the entries are to be printed.

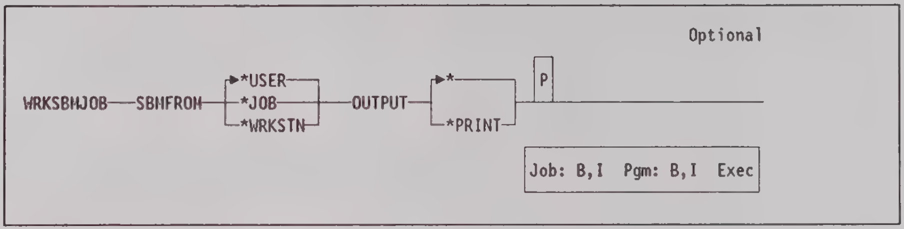
Example

```
WRKRPYLE   OUTPUT(*PRINT)
```

This command prints the entries in the system reply list.

WRKSBMJOB (Work with Submitted Jobs) Command

The Work with Submitted Jobs (WRKSBMJOB) command allows you to work with the status of all jobs submitted at a work station, in a job, or under a user profile. (Jobs submitted with DSPSBMJOB(*NO) specified on the Submit Job (SBMJOB), Submit Data Base Jobs (SBMDBJOB), or Submit Diskette Jobs (SBMDKTJOB) commands are not displayed by this command.)



SBMFROM Parameter

Specifies the type of submitted jobs that are displayed. Any jobs of that type that were submitted with the parameter DSPSBMJOB(*NO) specified on the SBMJOB, SBMDBJOB, or SBMDKTJOB commands are not included.

***USER:** The jobs displayed are those that were submitted from a job having the same user profile as the job in which this command is entered.

***JOB:** The jobs displayed are those that were submitted from the same job in which this command is entered.

***WRKSTN:** The jobs displayed are those that were submitted from the same work station at which this command is entered.

OUTPUT Parameter

Specifies if the display is shown that allows you to work with or change submitted jobs displayed at the requesting work station or printed with the job's spooled output. (Refer to Appendix A, "Expanded Parameter Descriptions" for an expanded description of the OUTPUT parameter, and to Appendix D, "Files Used by CL Commands" for the name of the printer file used by this command.)

***:** The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

***PRINT:** The output is printed with the job's spooled output.

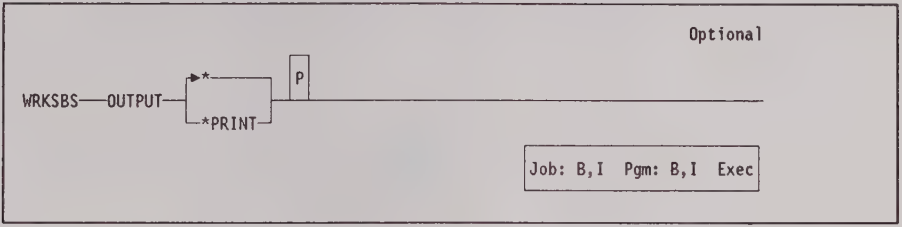
Example

```
WRKSBMJOB  SBMFROM(*USER)
```

This command allows you to work with or change a list of jobs that are submitted by a job running under the same user profile as the job where this command is run.

WRKSBS (Work with Subsystems) Command

The Work with Subsystem (WRKSBS) command allows you to work with each active subsystem in the system. Also, if one of the subsystems shown on the system display is selected, additional information listing all of the jobs active in that subsystem can be displayed.



OUTPUT Parameter

Specifies if the output is displayed at the requesting work station or printed with the job's spooled output. (Refer to Appendix A, "Expanded Parameter Descriptions" for an expanded description of the OUTPUT parameter, and to Appendix D, "Files Used by CL Commands" for the name of the printer file used by this command.)

***:** The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

***PRINT:** The output is printed with the job's spooled output.

Example

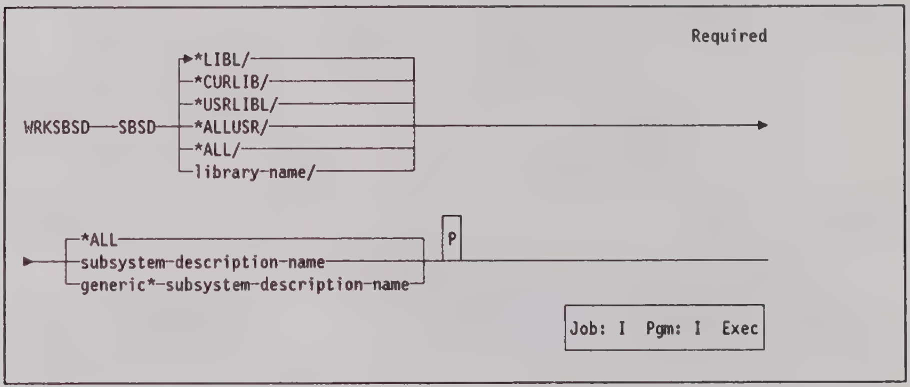
```
WRKSBS
```

This command, if entered from a work station, allows you to work with active subsystems. If the command is entered from a batch job, the information is directed to the job's output spooling queue being printed.

WRKSBSD (Work with Subsystem Description) Command

The Work with Subsystem Description (WRKSBSD) command allows you to work with a list of subsystem descriptions.

Restriction: You must have object operational and read authorities for the subsystem description before you can show its contents.



SBSD Parameter
Specifies the qualified name of the subsystem description that is shown.

The possible library values are:

***LIBL:** The library list is used to locate the subsystem description.

***CURLIB:** The current library for the job is used to locate the subsystem description. If no current library entry exists in the library list, QGPL is used.

***USRLIBL:** Only the libraries in the user portion of the job's library list are searched.

***ALLUSR:** All the nonsystem libraries, which include all user-defined libraries and the QGPL library, that not in the job's library list are searched. Libraries whose names start with the letter Q, other that QGPL, are not included.

***ALL:** All of the libraries in the system, including QSYS, are searched.

library-name: Specify the library where the subsystem description is located.

***ALL:** Specifies that all subsystem descriptions (in the libraries specified) are searched.

subsystem-description-name: Specify the name of the subsystem description being shown.

generic-subsystem-description-name:* Specify the generic name of the subsystem description being shown. For example, if you type T* in the library QGPL, you get a list of all the subsystem descriptions starting with T in library QGPL.

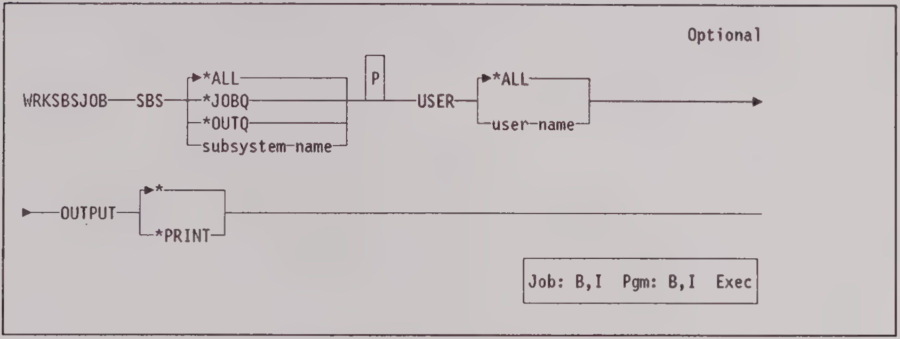
Example

```
WRKSBSD  SBSD(LIB6/ORDER*)
```

This command shows a list of all subsystem descriptions in library LIB6 that start with ORDER. From this list several SBSD related commands can be performed.

WRKSBSJOB (Work with Subsystem Jobs) Command

The Work with Subsystem Job (WRKSBSJOB) command allows you to work with jobs being processed by subsystems in the system, and of jobs that are on a job or output queue. If a user name is specified, only those jobs belonging to the specified user can be worked with or changed. Also, if one of the jobs shown on the subsystem display is selected, additional information about that job can be displayed.



SBS Parameter
Specifies the name of the subsystem (or all subsystems) for which the job name and job status of each job currently active in the subsystem, or jobs on a JOBQ or OUTQ, are worked with.

*ALL: All jobs in all subsystems have their job information displayed. In this case, jobs that are on job queues and on output queues are also worked with.

*JOBQ: Works with the jobs that are on a job queue.

*OUTQ: Works with the jobs that are on an output queue.

Note: This does not include system jobs or subsystem monitor jobs, but does include readers and writers.

subsystem-name: Specify the name of the subsystem. All active jobs in this subsystem are worked with.

USER Parameter
Specifies the name of the user whose jobs are worked with.

*ALL: All jobs being processed under all user names are worked with.

user-name: Specify a user name. All jobs with this user name are worked with.

OUTPUT Parameter
Specifies if the display is shown that allows you to work with a job displayed at the requesting work station or printed with the job's spooled output on a printer. (Refer to Appendix A, “Expanded Parameter Descriptions” for an expanded description of the OUTPUT parameter, and to Appendix D, “Files Used by CL Commands” for the name of the printer file used by this command.)

*: The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

*PRINT: The output is printed with the job's spooled output on a printer.

Examples

```
WRKSBSJOB
```

This command, entered from a work station, allows you to work with all jobs in all subsystems, and the jobs on the job queues and output queues. You can obtain additional information about one of the jobs by typing a 1 before the job's name.

```
WRKSBSJOB   SBS(QBATCH)  USER(JONES)
```

This command allows you to work with all jobs in the QBATCH subsystem that belong to the user profile of the user named JONES.

```
WRKSBSJOB   SBS(*OUTQ)
```

This command allows you to work with jobs that are on an output queue.

WRKSOC (Work with Sphere of Control) Command

The Work with Sphere of Control (WRKSOC) command allows you to change the sphere of control by adding or deleting control point names.

More information on sphere of control is in the *C & SM User's Guide*.

Restriction: This command is valid only in an interactive environment.

```
WRKSOC──────────────────────────────────────────────────────────────
                                                ┌──────────────────────┐
                                                │Job: I  Pgm: I  Exec  │
                                                └──────────────────────┘
```

There are no parameters for this command.

Example

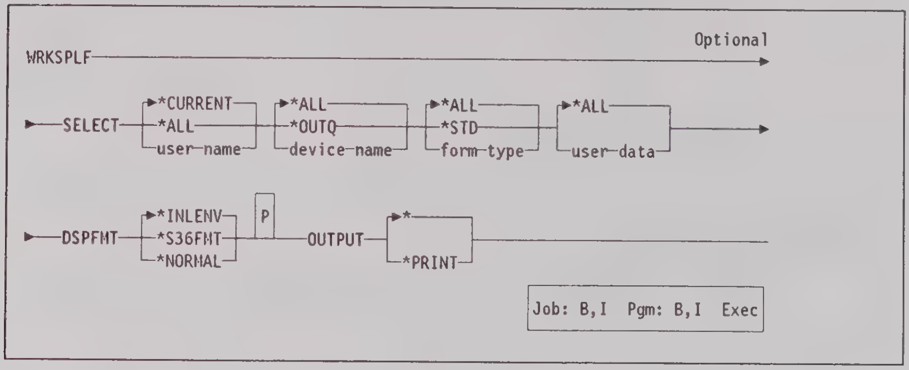
```
WRKSOC
```

This command shows the Work with Sphere of Control display.

WRKSPLF (Work with Spooled Files) Command

The Work with Spooled Files (WRKSPLF) command allows you to work with the spooled files on the system that meet the selection values specified by the SELECT keyword. You can display or print all the spooled files on the system or a selected subset of them.

Performance Considerations: The amount of time needed to show the list of spooled files is directly proportional to the number of spooled files in the list. On a system with a large number of spooled files, it may be necessary to use the SELECT parameter to view a subset of the system's spooled files.



SELECT Parameter

Specifies which group of files should be selected to be shown or printed. Four positional values can be specified to select the files; the user that created the file, the device that the file is queued for, the form type specified, or the user data tag associated with the file. Only files that meet each of the requirements are selected.

Possible values for user:

***CURRENT:** Only files created by the user running this command are selected.

***ALL:** Files created by all users are selected.

user-name: Only files created by the specified user are selected.

Possible values for device:

***ALL:** Files on the queue for any device or on any output queue are selected.

***OUTQ:** All files that are not queued for a device. These files are on output queues that are not associated with printers.

device-name: Only files on the queue for the specified device are selected.

Possible values for form type:

***ALL:** Files for all form types are selected.

***STD:** Only files which specify the standard form type are selected.

form-type: Only files with the specified form type are selected.

Possible values for user data:

***ALL:** Files with any user data tag specified are selected.

user-data: Only files with the specified user data tag are selected.

DSPFMT Parameter

Specifies the format and terminology to be used on the displays that result from running this command.

Note: If the System/36 display format is used, the OUTPUT parameter is ignored and the data is presented at the user's display station.

***INLENV:** The format and terminology that is used is determined by the SPCENV (special environment) value specified in your user profile. If the SPCENV value is *S36, the System/36 terminology is used. All other values result in AS/400 terminology being used.

***S36FMT:** System/36 terminology is used to display the information.

***NORMAL:** AS/400 system terminology is used to display the information.

OUTPUT Parameter

Specifies whether the output is shown at a display station or printed. The same information is presented in either case. If the output is shown, it is sent to the work station that entered the command.

*****:** The output is shown if requested by an interactive job or printed if requested by a batch job.

***PRINT:** The output is printed.

Example

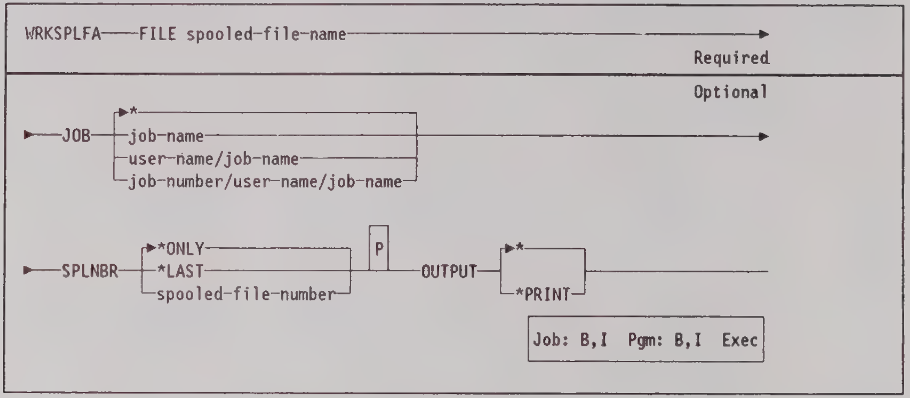
```
WRKSPLF  SELECT(*ALL *ALL *ALL *ALL)  DSPFMT(*NORMAL)
```

This command shows information about all the spooled files on the system. AS/400 system terminology is used to show the information. The information goes to the screen by default.

WRKSPLFA (Work with Spooled File Attributes) Command

The Work with Spooled File Attributes (WRKSPLFA) command allows you to work with the current attributes of the specified spooled output file.

The attributes can be shown after the file is opened and while its file entry is still on the output queue.



FILE Parameter
Specifies the name of the spooled output file to have its attributes shown.

JOB Parameter
Specifies the name of the job that created the spooled file.

***:** The job that entered this WRKSPLFA command is the job that created the spooled file.

job-name: Specify the name of the job that created the spooled file. If no job qualifier is given, all jobs currently in the system are searched for the simple job name. (For an expanded description of the JOB parameter and duplicate job names, see Appendix A, "Expanded Parameter Descriptions" in Volume 1.)

SPLNBR Parameter
Specifies the unique number of the spooled output file in the job whose attributes are being shown. (For an expanded description of the SPLNBR parameter, see Appendix A, "Expanded Parameter Descriptions" in Volume 1.)

***ONLY:** Only one spooled output file from the job has the specified file name; therefore, the number of the spooled file is not necessary.

***LAST:** The highest-numbered spooled output file with the specified file name is the file whose attributes are shown.

spooled-file-number: Specify the number of the spooled output file with the specified file name whose attributes are shown.

OUTPUT Parameter
Specifies whether the output is shown at a display station or printed. The same information is presented in either case. If the output is shown, it is sent to the work station that entered the command.

***:** The output is shown if requested by an interactive job or printed if requested by a batch job.

***PRINT:** The output is printed.

Example

```
WRKSPLFA  FILE(QPRINT)  OUTPUT(*PRINT)
```

This command sends a file containing the current attributes of the spooled output file QPRINT to the job's output queue to be printed. The job that entered this command must have produced only one output file named QPRINT.

WRKSSND (Work with Session Description) Command

The Work with Session Description (WRKSSND) command shows the current contents of the active or inactive Remote Job Entry (RJE) session description.

Restriction: To use this command, you must have read authority to the library where the session description is stored.

```
                   ┌►*LIBL/───────┐
 WRKSSND──SSND──┼─*CURLIB/───────┼─session-description-name──────────────►
                   └─library-name/─┘
                                                                    Required
─────────────────────────────────────────────────────────────────────────────
                                                                    Optional
 ►──OUTPUT──┬►*─────┬──────────────────────────────────────────────────────
             └─*PRINT─┘

                                                    ┌──────────────────────┐
                                                    │Job: B,I  Pgm: B,I  Exec│
                                                    └──────────────────────┘
```

SSND Parameter

Specifies the qualified name of the session description that is shown.

The possible library values are:

***LIBL:** The library list is used to locate the session description.

***CURLIB:** The current library for the job is used to locate the session description. If no current library entry exists in the library list, QGPL is used.

library-name: Specify the library where the session description is located.

OUTPUT Parameter

Specifies if the output from the command is shown at the requesting work station or printed with the spooled output of the job.

***:** The output is shown at the work station if it is requested by an interactive job. The output is printed with the spooled output of the job if it is requested by a batch job.

***PRINT:** The output is printed with the spooled output of the job. The IBM-supplied device file QPRTSSND prints the RJE session description.

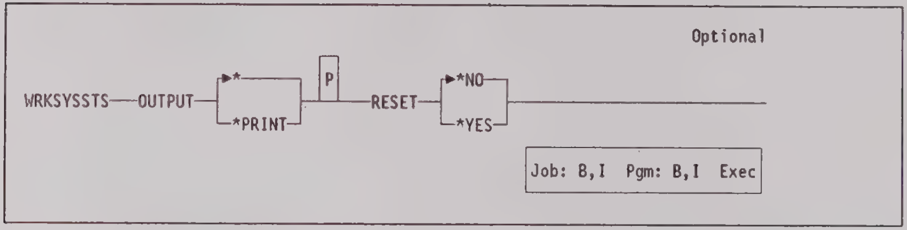
Example

```
WRKSSND  SSND(USERLIB/RJE)  OUTPUT(*)
```

This command (if entered from a batch job) sends a complete set of display information about the session description named RJE in library USERLIB to the job's spooling queue to print. If the command is entered from an interactive job, the session description menu is displayed, from which detailed selections can be made. The information includes the session attributes, communication entries, writer entries, and reader entries.

WRKSYSSTS (Work with System Status) Command

The Work with System Status (WRKSYSSTS) command allows you to work with information about the current status of the system. It displays the number of jobs currently in the system, the total capacity of system auxiliary storage pool (ASP) storage, the percentage of the system ASP storage currently in use, the amount of temporary storage currently in use, the maximum amount of temporary storage space required since the last initial program load (IPL), the percentage of machine addresses used, and statistical information related to each storage pool that currently has main storage allocated to it. The statistical information is gathered during an identified time interval (shown as the elapsed time); the data can either be updated by extending the measurement time interval, or it can be reset and a new time interval can be started to gather a new set of data.



OUTPUT Parameter

Specifies if the output is displayed at the requesting work station or printed with the job's spooled output on a printer. (Refer to Appendix A, "Expanded Parameter Descriptions" for an expanded description of the OUTPUT parameter, and to Appendix D, "Files Used by CL Commands" for the name of the printer file used by this command.)

***:** The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

***PRINT:** The output is printed with the job's spooled output on a printer.

RESET Parameter

Specifies if system status statistics fields are reset to zero, as if this was the first occurrence of the WRKSYSSTS command in this job.

***NO:** The system status statistics are not reset.

***YES:** The system status statistics are reset.

Example

```
WRKSYSSTS  OUTPUT(*PRINT)
```

This command directs the system status information to the job's output spooling queue for printing. If OUTPUT(*) is specified instead and the command was entered from a work station, the information about the system is displayed at the work station.

WRKTIE (Work with Technical Information Exchange) Command

The Work with Technical Information Exchange (WRKTIE) command shows the TIE main menu where you can select TIE tasks. These TIE tasks are:

- Send TIE files
- Receive TIE files
- Query TIE files

```
WRKTIE──SPTNUSRID support-network-user-ID──────────────────────────►

                                 ┌─┐
                                 │P│
SPTNPWD support-network-password─┴─┴───────────────────────────────►
                                                            Required
────────────────────────────────────────────────────────────────────
                                                            Optional
             ┌►*RTV──────────┐
►──ACCOUNT───┤               ├─────────────────────────────────────
             └account-number─┘

                                                ┌─────────────────┐
                                                │Job: I Pgm: I Exec│
                                                └─────────────────┘
```

SPTNUSRID Parameter
Specifies the user ID needed to sign on to the IBM Information Network.

SPTNPWD Parameter
Specifies the password needed to sign on to the IBM Information Network.

ACCOUNT Parameter
Specifies the network account number used to sign on to the IBM Information Network. If the account number is not specified, the account from the contact data base is used.

***RTV:** The account number from the contact data base is used (refer to the WRCNTINF command).

account-number: Specify account number that is to be used.

Example

```
WRKTIE  SPTNUSRID(ACME)   SPTNPWD(11111)  ACCOUNT(11420880)
```

This command displays the TIE main menu.

WRKTXTPRF (Work with Text Profiles) Command

The Work with Text Profiles (WRKTXTPRF) command shows the Work with Text Profiles display. From this display, the user can create, revise, copy, or delete profiles that provide default values when using the word processing function of the office product.

More information on text profiles is in the *Word Processing User's Guide*.

```
                                                          Optional
WRKTXTPRF─────────────────────────────────────────────────────────

                                               ┌──────────────────────┐
                                               │Job: I  Pgm: I  Exec  │
                                               └──────────────────────┘
```

There are no parameters for this command.

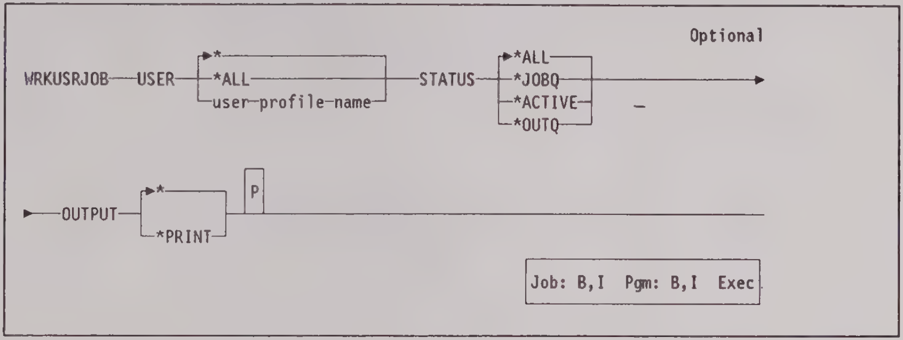
Example

```
WRKTXTPRF
```

This command shows the Work with Text Profiles display.

WRKUSRJOB (Work with User Jobs) Command

The Work with User Jobs (WRKUSRJOB) command provides a list of selected user jobs.



USER Parameter

This parameter allows you to work with user jobs defined by the user profile of the job. User jobs include interactive jobs, submitted batch jobs, communications-evoked batch jobs, MRT batch jobs, and autostart jobs. User jobs do not include system jobs, subsystem monitor jobs, spooling readers, or spooling writers.

***:** Specifies user jobs with the current user profile are worked with.

***ALL:** Specifies user jobs with all user profiles are worked with.

user-profile name: Specify user jobs with a specified user profile are worked with.

STATUS Parameter

Specifies the status of the user jobs being worked with.

***ALL:** All statuses of user jobs can be worked with, including jobs on job queues, active jobs, and jobs on output queues.

***JOBQ:** Specifies only user jobs that are on job queues are worked with.

***ACTIVE:** Specifies only user jobs that are active are worked with, including all group jobs and system request jobs.

***OUTQ:** Specifies only user jobs that have completed running but still have output on an output queue are worked with.

OUTPUT Parameter

Specifies if the output is displayed at the requesting work station or printed with the job's spooled output on a printer. (Refer to Appendix A, "Expanded Parameter Descriptions" for an expanded description of the OUTPUT parameter, and to Appendix D, "Files Used by CL Commands" for the name of the printer file used by this command.)

***:** The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

***PRINT:** The output is printed with the job's spooled output on a printer.

Examples

```
WRKUSRJOB  USER(DICK)
```

This command allows you to work with a list of jobs that are running with a user profile name of DICK.

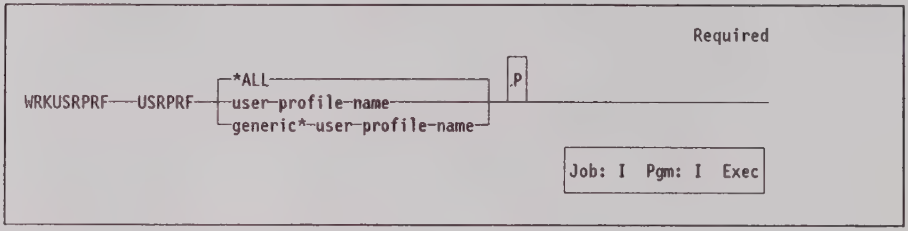
```
WRKUSRJOB  USER(*ALL)  STATUS(*JOBQ)
```

This command allows you to work with a list of all jobs on job queues.

WRKUSRPRF (Work with User Profiles) Command

The Work with User Profiles (WRKUSRPRF) command shows a list of user profiles from which you can perform several related functions.

Restriction: You must have object management authority or ownership of the user profile to display or change the user profile.



USRPRF Parameter

Specifies the name or generic name of the user profile(s) you want work with.

***ALL:** Specifies a list of all the user profiles that you own or have authority to see.

user-profile-name: Specifies the name of the user profile that you want to work with.

generic-user-profile-name:* Specifies the generic name of the user profile that you want to work with. A generic name can be specified as a character string that contains one or more characters followed by an asterisk (*). If a generic user profile name is specified, then all user profiles that have names with the same prefix as the generic user profile name and that the user has some authority for are displayed.

Example

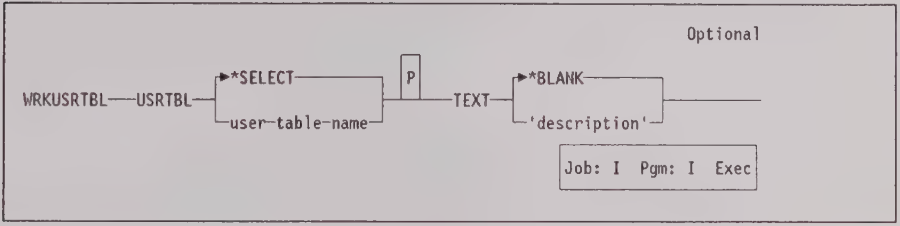
```
WRKUSRPRF (RO*)
```

This command displays a list of user profiles starting with the letters RO.

WRKUSRTBL (Work with User Table) Command

The Work with User Table (WRKUSRTBL) command lets you create finance user tables and, once they are created, lets you add or delete user IDs in these tables. Several finance user tables can be defined, but you must use unique names for each table. If the user table you specify on this command does not exist, it is created.

User table updates can be accessed by any finance job that is submitted after all changes are completed.



USRTBL Parameter

Specifies the name of a table that contains finance user IDs.

***SELECT:** Displays the list of existing user table entries. Through this display you can add new user table names to the system, delete old user tables from the system, or modify the contents of existing user tables.

user-table-name: Specifies the name of the user table that you want to work with.

TEXT Parameter

Specifies the user-defined text that describes the user table. (For an expanded description of the TEXT parameter, see Appendix A, "Expanded Parameter Descriptions.")

***BLANK:** No text is specified.

'description': Specify no more than 50 characters, enclosed in apostrophes. If you enter text in this field and specify USRTBL(*SELECT), the text you enter will be carried over to the Work with User Table display.

Examples

```
WRKUSRTBL USRTBL(*SELECT)
```

This command shows the names of all finance user tables. You can add a new table, select an existing table for update, or remove tables on the Work with User Table display.

```
WRKUSRTBL USRTBL(USRTBL1) TEXT('User group one')
```

This command shows the user table USRTBL1, allowing you to add, change, or remove table entries. If USRTBL1 does not exist when the WRKUSRTBL command is entered, a new user table is created using the specified text.

WRKWTR (Work with Writers) Command

The Work with Writers (WRKWTR) command allows you to work with the status of printers and writers. This can be the overall status of all writers or all printers along with writer information for the printers, or detailed status of a specific writer. This status information may be directed to a display station or a printer. The status of the writers may change while the command is running.

```
                                                         Optional
               ┌►*PRT──────────┐
WRKWTR───WTR───┼──*ALL─────────┼─────────────────────────────────►
               └──writer-name──┘

       (1)      ┌►*INLENV─┐  ┌─┐    (2)      ┌►*──────┐
►───DSPFMT──────┼──*S36FMT─┼──│P│───OUTPUT───┤        ├──────────────
                └──*NORMAL─┘  └─┘             └─*PRINT─┘

(1) This parameter is valid only if WTR(*PRT) is specified.
(2) This parameter is ignored if DSPFMT(*S36FMT) is specified.  The
    data will be presented at your terminal.

                                         ┌──────────────────────────┐
                                         │Job: B,I  Pgm: B,I  Exec │
                                         └──────────────────────────┘
```

WTR Parameter

Specifies the name of the spooling writer that detailed information is displayed for, or specifies that the main attributes and status of *all* spooling writers or all printers are being shown.

<u>*PRT</u>: All printers configured on the system are shown along with information about the writers started for them.

***ALL:** The attributes and the current status of all spooling writers are shown.

writer-name: Specify the name of the spooling writer that the detailed description is shown for.

DSPFMT Parameter

Specifies the format and terminology to be used on the displays that result from entering this command.

<u>*INLENV</u>: The format and terminology to be used is determined by the special environment value (SPCENV) specified in your user profile. If the SPCENV value is *S36, the System/36 terminology is used. All other values result in AS/400 terminology being used.

***S36FMT:** System/36 terminology is used to present the information.

***NORMAL:** AS/400 system terminology is used to present the information.

OUTPUT Parameter

Specifies whether the output from the command is shown at the requesting work station or printed with the job's spooled output on a printer. (Refer to Appendix A, "Expanded Parameter Descriptions" in Volume 1 for an expanded description of the OUTPUT parameter, and to Appendix D, "Files Used by CL Commands" in Volume 1 for the name of the printer file used by this command.)

<u>*</u>: The output is shown (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

***PRINT:** The output is printed with the job's spooled output on a printer.

Examples

```
WRKWTR
```

This command shows a list of all printers and information about the writers started to them.

```
WRKWTR  WTR(DISKWRITE)
```

This command shows detailed information about the writer DISKWRITE.

AS/400™ Programming:
Control Language Reference
Volume 5

SC21-9779-1

READER'S COMMENT FORM

Please use this form only to identify publication errors or to request changes in publications. Direct any requests for additional publications, technical questions about IBM systems, changes in IBM programming support, and so on, to your IBM representative or to your IBM-approved remarketer. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

☐ If your comment does not need a reply (for example, pointing out a typing error), check this box and do not include your name and address below. If your comment is applicable, we will include it in the next revision of the manual.

☐ If you would like a reply, check this box. Be sure to print your name and address below.

Page number(s): Comment(s):

Please contact your IBM representative or your IBM-approved remarketer to request additional publications.

Name

Company or Organization

Address

City State Zip Code

Phone No.

Area Code

No postage necessary if mailed in the U.S.A.

Fold and Tape **Please do not staple**

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS / PERMIT NO. 40 /ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Information Development
Department 245
Rochester, Minnesota, U.S.A. 55901

Fold and Tape **Please do not staple**

IBM®

AS/400™ Programming:
Control Language Reference
Volume 5

SC21-9779-1

READER'S COMMENT FORM

Please use this form only to identify publication errors or to request changes in publications. Direct any requests for additional publications, technical questions about IBM systems, changes in IBM programming support, and so on, to your IBM representative or to your IBM-approved remarketer. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

☐ If your comment does not need a reply (for example, pointing out a typing error), check this box and do not include your name and address below. If your comment is applicable, we will include it in the next revision of the manual.

☐ If you would like a reply, check this box. Be sure to print your name and address below.

Page number(s): Comment(s):

Please contact your IBM representative or your IBM-approved remarketer to request additional publications.

Name ______________________

Company or Organization ______________________

Address ______________________

____
City State Zip Code

Phone No. ______________________
Area Code

No postage necessary if mailed in the U.S.A.

Fold and Tape **Please do not staple**

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS / PERMIT NO. 40 /ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Information Development
Department 245
Rochester, Minnesota, U.S.A. 55901

Fold and Tape **Please do not staple**

IBM®

AS/400™ Programming:
Control Language Reference
Volume 5

SC21-9779-1

READER'S COMMENT FORM

Please use this form only to identify publication errors or to request changes in publications. Direct any requests for additional publications, technical questions about IBM systems, changes in IBM programming support, and so on, to your IBM representative or to your IBM-approved remarketer. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

☐ If your comment does not need a reply (for example, pointing out a typing error), check this box and do not include your name and address below. If your comment is applicable, we will include it in the next revision of the manual.

☐ If you would like a reply, check this box. Be sure to print your name and address below.

Page number(s): Comment(s):

Please contact your IBM representative or your IBM-approved remarketer to request additional publications.

Name

Company or Organization

Address

City State Zip Code

Phone No.

Area Code

No postage necessary if mailed in the U.S.A.

Fold and Tape **Please do not staple**

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS / PERMIT NO. 40 /ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Information Development
Department 245
Rochester, Minnesota, U.S.A. 55901

Fold and Tape **Please do not staple**

IBM®

Program Number
5728-SS1

21F2719

SC21-9779-1

Printed in U.S.A.

93X0313

SX21-9925-00

Printed in U.S.A.